Why ADAGRAD Fails for Online Topic Modeling

You Lu
Computer Science (CS)
University of Colorado Boulder
Boulder, CO
you.lu@colorado.edu

Jeffrey Lund
Computer Science (CS)
Brigham Young University
Provo, UT
jefflund@byu.edu

Jordan Boyd-Graber
CS, iSchool, LSC, and UMIACS
University of Maryland
College Park, MD
jbg@umiacs.umd.edu

Abstract

Online topic modeling, i.e., topic modeling with stochastic variational inference, is a powerful and efficient technique for analyzing large datasets, and ADAGRAD is a widely-used technique for tuning learning rates during online gradient optimization. However, these two techniques do not work well together. We show that this is because ADAGRAD uses accumulation of previous gradients as the learning rates' denominators. For online topic modeling, the magnitude of gradients is very large. It causes learning rates to shrink very quickly, so the parameters cannot fully converge until the training ends.

Probabilistic topic models (Blei, 2012) are popular algorithms for uncovering hidden thematic structure in text. They have been widely used to help people understand and navigate document collections (Blei et al., 2003), multilingual collections (Hu et al., 2014), images (Chong et al., 2009), networks (Chang and Blei, 2009; Yang et al., 2016), etc. Probabilistic topic modeling usually requires computing a posterior distribution over thousands or millions of latent variables, which is often intractable. Variational inference (Blei et al., 2016, VI) approximates posterior distributions. Stochastic variational inference (Hoffman et al., 2013, SVI) is its natural online extension and enables the analysis of large datasets.

Online topic models (Hoffman et al., 2010; Bryant and Sudderth, 2012; Paisley et al., 2015) optimize the global parameters of interest using stochastic gradient ascent. At each iteration, they sample data points to estimate the gradient. In practice, the sample has only a small percentage of the vocabulary. The resulting sparse gradients hurt performance. ADAGRAD (Duchi et al., 2011) is designed for high dimensional online optimization problems and adjusts learning rates for each dimension, favoring rare features. This makes ADAGRAD well-suited for tasks with sparse gradients such as distributed deep networks (Dean et al., 2012), forward-backward splitting (Duchi and Singer, 2009), and regularized dual averaging methods (Xiao, 2010).

Thus, it may seem reasonable to apply ADAGRAD to optimize online topic models. However, ADAGRAD is not suitable for online topic models (Section 1). This is because to get a topic model, the training algorithm must break the symmetry between parameters of words that are highly related to the topic and words that are not related to the topic. Before the algorithm converges, the magnitude of gradients of the parameters are very large. Since ADAGRAD uses the accumulation of previous gradients as learning rates' denominators, the learning rates shrink very quickly. Thus, the algorithm cannot break the symmetry quickly. We provide solutions for this problem. Two alternative learning rate methods, i.e., ADADELTA (Zeiler, 2012) and ADAM (Kingma and Ba, 2014), can address this incompatibility with online topic models. When the dataset is small enough, e.g., a corpus with only hundreds of documents, ADAGRAD can still work.

1 Buridan's Optimizer

Latent Dirichlet allocation (Blei et al., 2003, LDA) is perhaps the most well known topic model. In this section, we analyze problems with ADAGRAD for online LDA (Hoffman et al., 2010), and provide some solutions. Our analysis is easy to generalize to other online topic models, e.g., online Hierarchical Dirichlet Process (Wang et al., 2011, HDP).

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 455–460
Copenhagen, Denmark, September 7–11, 2017.

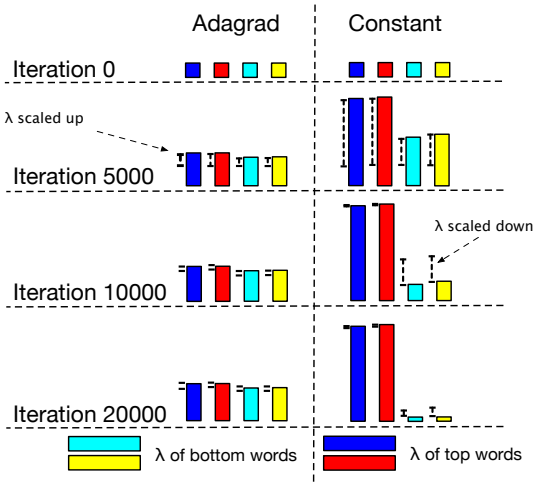

Figure 1: Illustration of ADAGRAD's problem. Initially, the topic does not favor particular words over others, so the training algorithm incorrectly increases the parameters of bottom words. Then, ADAGRAD learning rates decrease too quickly, leaving the tie between top and bottom unbroken. Thus, the algorithm fails to form appropriate topics. A constant rate easily breaks the tie. When the tie is broken, the algorithm decreases the parameters of bottom words and increases the parameters of top words until convergence.

1.1 Online LDA

To train LDA, we want to compute the posterior

$$p(\beta,\theta,z \mid w,\alpha,\eta) \propto \prod_{k=1}^{K} p(\beta_k \mid \eta) \cdot \prod_{d=1}^{D} p(\theta_d \mid \alpha) \prod_{n=1}^{N_d} p(z_{dn} \mid \theta_d) p(w_{dn} \mid \beta_{z_{dn}}),$$

where β_k is the topic-word distribution for the k^{th} of K topics, θ_d is the document-topic distribution for the d^{th} of D document, z_{dn} is the topic assignment for the n^{th} of N_d words in in the d^{th} document, w_{dn} is the word type of the n^{th} word in the d^{th} document, with α and η the Dirichlet priors over the document-topic and topic-word distributions.

However, this is intractable. Stochastic variational inference (SVI) is a popular approach for approximation. It first posits a mean field variational distribution

$$q(\beta,\theta,z \mid \lambda,\gamma,\phi) = \prod_{k=1}^{K} q(\beta_k \mid \lambda_k) \cdot \prod_{d=1}^{D} q(\theta_d \mid \gamma_d) \prod_{n=1}^{N_d} q(z_{dn} \mid \phi_{dn}),$$

where γ (Dirichlet) and ϕ (multinomial) are local parameters and λ (Dirichlet) is a global parameter. SVI then optimizes the variational parameters to minimize the KL divergence between the variational distribution and the true posterior.

At iteration t, SVI samples a document d from the corpus and updates the local parameters:

$$\phi_{vk}^{d} \propto \exp\left\{\Psi\left(\gamma_{dk}\right) + \Psi\left(\lambda_{kv}^{(t)}\right) - \Psi\left(\sum_i \lambda_{ki}^{(t)}\right)\right\}, \quad (1)$$

$$\gamma_k^{(t)} = \alpha + \sum_v n_v \phi_{vk}^{d}, \quad (2)$$

where n_v is the number of words v in d, and $\Psi(.)$ is the digamma function. After finding ϕ^d and γ^d, SVI optimizes the global parameters using stochastic gradient ascent,

$$\begin{aligned} \lambda_{kv}^{(t+1)} &= (1-\rho_{kv}^{(t)})\lambda_{kv}^{(t)} + \rho_{kv}^{(t)}(\eta + D\phi_{vk}^{d} n_{dv}) \\ &= (1-\rho_{kv}^{(t)})\lambda_{kv}^{(t)} + \rho_{kv}^{(t)}\hat{\lambda}_{kv}^{(t)} \\ &= \lambda_{kv}^{(t)} + \rho_{kv}^{(t)} g_{kv}^{(t)}, \end{aligned} \quad (3)$$

where $\rho^{(t)}$ is the learning rate, $\hat{\lambda}_{kv}^{(t)} = \eta + D\phi_{vk}^{d} n_{dv}$ is the intermediate parameter and $g_{kv}^{(t)} = -\lambda_{kv}^{(t)} + \hat{\lambda}_{kv}^{(t)}$ is the gradient.

1.2 ADAGRAD for Online LDA

In general, $\rho_{kv}^{(t)} = \kappa^{(t)}$, for all $v \in 1,..,V$ and $k \in 1,...,K$, where $\kappa^{(t)}$ can be a decreasing rate (Hoffman et al., 2013), a small constant (Collobert et al., 2011) or an adaptive rate (Ranganath et al., 2013). These three methods are all global learning rate methods, which cannot adaptively adjust learning rate for each dimension of the parameter, or address the problems caused by sparse gradients.

ADAGRAD is a popular learning rate method designed for online optimization problems with high dimension and sparse gradients. Thus, it seems reasonable to apply ADAGRAD to update learning rates for online topic models. When using ADAGRAD (Duchi et al., 2011) with online LDA, the update rule for the each learning rate is

$$\rho_{kv}^{(t)} = \frac{\rho_0}{\sqrt{\epsilon + \sum_{i=0}^{t}\left(g_{kv}^{(i)}\right)^2}}, \quad (4)$$

where ρ_0 is a constant, and a very small ϵ guarantees that the learning rates are non-zero.

1.3 ADAGRAD's Indecision

A philosophical thought experiment provides us with the story of Buridan's ass (Bayle, 1826): situated between two piles of equally tasty hay, the poor animal starved to death. ADAGRAD faces a similar problem in breaking the symmetries of common variational inference initializations. For convenience, we unfold an example with a single document at each iteration. Our analysis generalizes to mini-batches.

Initially, the topics $\beta_{1:K}$ do not favor particular words over others as inference cannot know *a priori* which words will have high probability in a particular topic. The algorithm must break ties between parameters of the top and bottom words in a topic. Unfortunately, the momentum of ADAGRAD fails for topic models. We now explain why this is.

ADAGRAD looks to the gradient for clues about what features will be important. This is because before the equilibrium is broken, the values of different λ_{kv} are close, so Equation 1 will be approximately seen as $\phi^d_{vk} \propto \exp\{\Psi(\gamma_{dk})\}$, which implicates that λ has very small influence on the optimization of ϕ. If some topics are prevalent in the sampled document d, large probability will be assigned to the corresponding $\phi_{\cdot k}$, meaning that all words in document d are treated as top words. The initial clues are at best random and at words counter productive.

However, ADAGRAD uses these cues to prefer some dimensions over others. Let λ^* be the optimum; the topic ADAGRAD should find at convergence: $\lambda^*_{kv} \approx \mathbb{E}\left[\hat{\lambda}^{(t)}_{kv}\right]$. By definition, once the algorithm converges, λ^*_{kv} for top words will have very large values while λ^*_{kv} for bottom words will be small. After using noisy momentum terms, it must overcome initial faulty signals.

We now show the lower and upper bounds of $\mathbb{E}\left[\hat{\lambda}^{(t)}_{kv}\right]$ to show how big of an uphill battle ADAGRAD faces. Expanding the update rule,

$$\begin{aligned}\mathbb{E}\left[\hat{\lambda}^{(t)}_{kv}\right] &= \mathbb{E}\left[\eta + D\phi^d_{vk} n_{dv}\right] \\ &= \eta + D\bar{n}_v \mathbb{E}\left[\phi_{vk}\right],\end{aligned}$$

where $\bar{n}_v = \sum_{i=1}^{D} n_{iv}/D$, and ϕ_{vk} is the probability that word v is assigned to topic k. For a bottom word, $\phi_{vk} \to 0$. For a top word, $\phi_{vk} \geq 1/K$. After convergence, for a bottom word $\mathbb{E}\left[\phi_{vk}\right] \approx \eta$. For a top word, $1/K \leq \mathbb{E}\left[\phi_{vk}\right] \leq 1$. Thus, the lower and upper bounds of $\mathbb{E}\left[\hat{\lambda}^{(t)}_{kv}\right]$ are

$$\eta + (1/K)D\bar{n}_v \leq \mathbb{E}\left[\hat{\lambda}^{(t)}_{kv}\right] \leq \eta + D\bar{n}_v.$$

For a large datasets, $D\bar{n}_v$ should be large. Thus for top words, λ^*_{kv} will converge to a large value: quite a large hill to climb.

How quickly the algorithm climbs the hill is inversely proportional to the gradient size. We next show that the magnitude of gradients of top words are very large before the algorithm converges. Let g^* be the gradient after convergence. We show the bounds of $|g_{kv}|$, where $|.|$ is the absolute value, in the following:

$$\begin{aligned}|\, g^*_{kv}\,| &= |\, -\lambda^*_{kv} + \eta + D\phi^d_{vk} n_{dv}\,| \\ &\approx |\, -\eta - D\bar{n}_v E[\phi_{vk}] + \eta + D\phi^d_{vk} n_{dv}\,| \\ &\approx \mathbb{E}\left[\phi_{vk}\right] * D\,|\, n_{dv} - \bar{n}_v\,|\,.\end{aligned}$$

Thus,

$$(D/K)\,|\, n_{dv} - \bar{n}_v\,| \;\leq\; |\, g^*_{kv}\,| \;\leq D\,|\, n_{dv} - \bar{n}_v\,|\,.$$

Only when $n_{dv} = \bar{n}_v$, does $|\, g^{(t)}_{kv}\,| = 0$. Otherwise, due to the large D, $|\, g^*_{kv}\,|$ will be large. However, in practice, n_{dv} varies largely from document to document, which leads to large values of $|\, g^*_{kv}\,|$. Based on the gradient's property, when λ_{kv} is far away from the optimum, $|\, g^{(t)}_{kv}\,| \geq |\, g^*_{kv}\,|$. Thus, the values of $|\, g^{(t)}_{kv}\,|$ for the top words are very large before convergence.

ADAGRAD uses the accumulations of previous gradients as learning rates' denominators. Because of these large gradients in the first several iterations, learning rates soon decrease to small values; even if a topic has gathered a few words, ADAGRAD lacks the momentum to move other words into the topic. These small learning rates slows the updates of λ.

In sum, the initial gradient signals confuse the algorithm, the gradients are large enough to impede progress later, and large datasets imply a very large hill the algorithm must climb. Since the update progresses slowly, online LDA needs more iterations to break the equilibrium. Because the gradients of all words are still very large, the learning rates decrease quickly, which makes the update progress slower. When the update progresses more slowly, online LDA needs more iterations to break the tie. This cycle repeats, until some learning rates decrease to zero and learning effectively stops. Thus, the algorithm will never break the tie or infer good topics. Figure 1 illustrates the problem of online LDA with ADAGRAD.

1.4 Alternative Solutions

ADADELTA (Zeiler, 2012) and ADAM (Kingma and Ba, 2014) are extensions to ADAGRAD. ADADELTA does not have guaranteed convergence on convex optimization problems. Even though ADAM has a theoretical bound on its convergence rate, it is controlled by and sensitive to several learning rate parameters. For good performance with ADAM, manual adjustment is necessary. In addition, since ADADELTA computes the moving average of updates, and ADAM needs to compute the bias-corrected gradient estimate, they require more intricate implementations. Consequently, these two methods are not as popular as ADAGRAD for beginners. However, for SVI latent variable models, they can address the problems with ADAGRAD.

ADADELTA updates the learning rates with the following rule:

$$\rho_{kv}^{(t)} = \frac{\sqrt{\mathbb{E}\left[(\lambda_{kv}^{(t)} - \lambda_{kv}^{(t-1)})\right] + \varepsilon}}{\sqrt{\mathbb{E}\left[g_{kv}^{(t)}\right] + \varepsilon}}, \quad (5)$$

where $\mathbb{E}\left[x^{(t)}\right] = \rho_0 \mathbb{E}\left[x^{(t-1)}\right] + (1-\rho_0)(x^{(t)})^2$, ρ_0 is a decay constant, and ε is for numerical stability.

ADAM's update rule is determined based on estimates of first and second moments of the gradients:

$$\begin{aligned} m_{kv}^{(t)} &= b_m m_{kv}^{(t-1)} + (1-b_m) g_{kv}^{(t)}, \\ u_{kv}^{(t)} &= b_u u_{kv}^{(t-1)} + (1-b_u)(g_{kv}^{(t)})^2, \\ \hat{m}_{kv}^{(t)} &= \frac{m_{kv}^{(t)}}{1-b_m^t}, \hat{u}_{kv}^{(t)} = \frac{u_{kv}^{(t)}}{1-b_u^t}, \\ \lambda_{kv}^{(t+1)} &= \lambda_{kv}^{(t)} + \rho_0 \hat{m}_{kv}^{(t)} / (\sqrt{\hat{u}_{kv}^{(t)}} + \varepsilon), \quad (6) \end{aligned}$$

where ρ_0 is a constant, b controls the decay rate.

Both ADADELTA and ADAM use the moving average of gradients as the denominator of learning rates. The learning rates will not monotonically decrease, but vary in a certain range. This property prevents online topic models from being trapped and breaks the tie between top words and bottom topic words. ADAM in particular uses bias-corrected estimate of gradient $\hat{m}_{kv}$, rather than the original stochastic gradient g_{kv} to guide direction for the optimization and therefore achieves better results.

In addition, the magnitude of gradients is proportional to the dataset's size. Thus, when the dataset is small enough, ADAGRAD will still work.

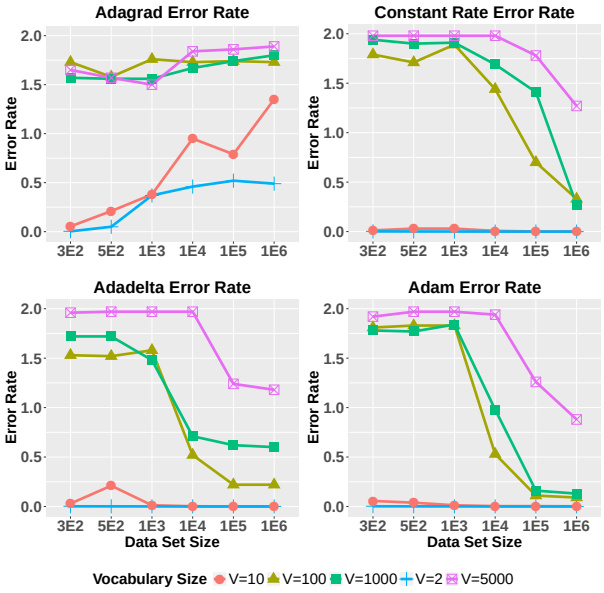

Figure 2: Experimental results on synthetic data sets. We vary the vocabulary size V, and the number of documents D. ADADELTA, ADAM and constant rate perform better with more data, while ADAGRAD only does well with small values of D.

2 Empirical Study

We study three datasets: **synthetic data**, **Wikipedia** and **SMS spam corpus**.[1] We use the generative process of LDA to generate synthetic data. We vary the vocabulary size $V \in \{2, 10, 100, 1000, 5000\}$, and the number of documents $D \in \{300, 500, 10^3, 10^4, 10^5, 10^6\}$. The **Wikipedia** dataset consists of 1M articles collected from Wikipedia.[2] The vocabulary is the same as (Hoffman et al., 2010). The SMS corpus is a small corpus containing 1084 documents.

2.1 Metrics and Settings

Error rate: For experiments on synthetic data set, we use error rate

$$\text{Error}(\hat{\beta}) = \frac{1}{K} \sum_{k=1}^{K} \min_i ||\hat{\beta}_i - \beta_k||_1 \quad (7)$$

to measure the difference between the estimated $\hat{\beta}$ and the known β. The min greedily matches each $\hat{\beta}_k$ to its best fit. While an uncommon metric for unsupervised algorithms, on the synthetic data we have the true β.

[1] http://www.esp.uem.es/jmgomez/smsspamcorpus/
[2] http://www.wikipedia.org

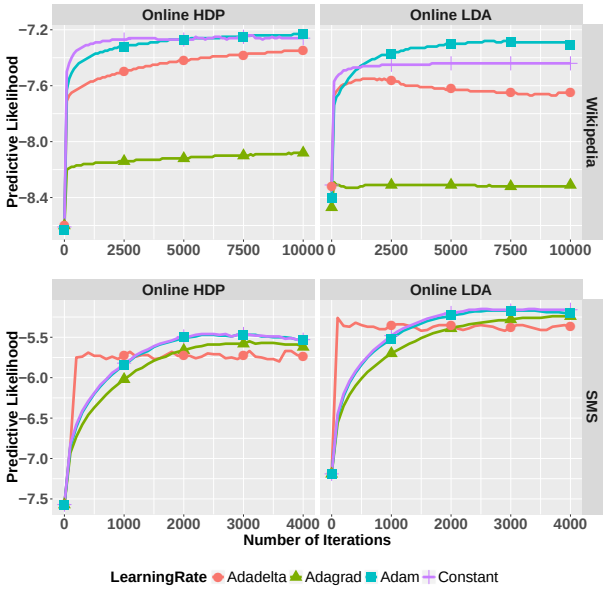

Figure 3: Experimental results on real corpora. Larger predictive likelihood is better. On Wikipedia, ADAGRAD has does worse than other methods. On SMS corpus, ADAGRAD is competitive.

Predictive likelihood: For experiments on real data sets, we use per-word likelihood (Hoffman et al., 2013) to evaluate the model quality. We randomly hold out 10K documents and 100 documents on Wikipedia and SMS respectively.

Settings: In the experiments on synthetic data, we use online LDA (Hoffman et al., 2010), since the data is generated by LDA. In the experiments on real datasets, we use online LDA and online HDP (Wang et al., 2011). In the experiments on Wikipedia, we set the number of topics $K = 100$ and the mini-batch size $M = 100$. In the experiments on SMS corpus, we set $K = 10$ and $M = 20$. For ADAM, we use the default setting of b, and set $\rho_0 = 10$ and $\epsilon = 1000$. For ADADELTA, we set $\epsilon = 1000$. For ADAGRAD, we set $\rho_0 = \epsilon = 1$. These are best settings for these three methods. The best constant rate is 10^{-3}.

2.2 Experimental Results

Figure 2 illustrates the experimental results on synthetic datasets. ADAGRAD only works well with small datasets. When the number of documents increases, ADAGRAD performance degrades. Conversely, other methods can handle more documents.

Figure 3 illustrates experimental results on real corpora. ADAGRAD gets competitive results to the other algorithms on the small SMS corpus. However on very large Wikipedia corpus, ADAGRAD fails to infer good topics, and its predictive ability is worse than the other methods. While ADADELTA and ADAM work well on Wikipedia, ADAM is the clear winner between the two.

3 Conclusion

ADAGRAD is a simple and popular technique for online learning, but is not compatible with traditional initializations and objective functions for online topic models. We show that practitioners are best off using simpler online learning techniques or ADADELTA and ADAM, which are two variants of ADAGRAD, which use the moving average of gradients as denominator. These two methods avoid ADAGRAD's problem. In particular, ADAM performs much better for prediction.

We would like to build a deeper understanding of which aspects of an unsupervised objective, near-uniform initialization, and non-identifiability contribute to these issues and to discover other learning problems that may share these issues.

Acknowledgments

We thank the anonymous reviewers, Stephan Mandt, Alp Kucukelbir, Bill Foland, Forough Poursabzi-Sangdeh and Alvin Grissom II for their insightful comments. Boyd-Graber and Lu's contribution is supported by NSF grants NCSE-1422492 and IIS-1409287, (UMD). Boyd-Graber is also supported by IIS-1564275 and IIS-1652666. Lund is supported by collaborative NSF Grant IIS-1409739 (BYU). Any opinions, findings, results, or recommendations expressed here are of the authors and do not necessarily reflect the view of the sponsor.

References

Pierre Bayle. 1826. *An historical and critical dictionary, selected and abridged*. Number 1 in An historical and critical dictionary, selected and abridged. https://books.google.com/books?id=cDsN3xOyOoC.

David M Blei. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2016. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670* .

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Michael Bryant and Erik B Sudderth. 2012. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Proceedings of Advances in Neural Information Processing Systems*.

Jonathan Chang and David M Blei. 2009. Relational topic models for document networks. In *Proceedings of Artificial Intelligence and Statistics*. volume 9, pages 81–88.

Wang Chong, David Blei, and Fei-Fei Li. 2009. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition*. IEEE.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. 2012. Large scale distributed deep networks. In *Proceedings of Advances in Neural Information Processing Systems*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10:2899–2934.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent Dirichlet allocation. In *Proceedings of Advances in Neural Information Processing Systems*.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research* 14:1303–1347.

Yuening Hu, Ke Zhai, Vlad Eidelman, and Jordan Boyd-Graber. 2014. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of the Association for Computational Linguistics*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

John Paisley, Chong Wang, David M Blei, and Michael I Jordan. 2015. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37:256–270.

Rajesh Ranganath, Chong Wang, David M Blei, and Eric P Xing. 2013. An adaptive learning rate for stochastic variational inference. In *Proceedings of the International Conference of Machine Learning*.

Chong Wang, John William Paisley, and David M Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of Artificial Intelligence and Statistics*.

Lin Xiao. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research* 11:2543–2596.

Weiwei Yang, Jordan Boyd-Graber, and Philip Resnik. 2016. A discriminative topic model using document network structure. In *Association for Computational Linguistics*.

Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .