Particle Filter Rejuvenation and Latent Dirichlet Allocation

Chandler May,[†] **Alex Clemmer**[‡] and **Benjamin Van Durme**[†]
[†]Human Language Technology Center of Excellence
Johns Hopkins University
[‡]Microsoft

cjmay@jhu.edu, clemmer.alexander@gmail.com, vandurme@cs.jhu.edu

Abstract

Previous research has established several methods of *online* learning for latent Dirichlet allocation (LDA). However, *streaming* learning for LDA—allowing only one pass over the data and constant storage complexity—is not as well explored. We use reservoir sampling to reduce the storage complexity of a previously-studied online algorithm, namely the particle filter, to constant. We then show that a simpler particle filter implementation performs just as well, and that the quality of the initialization dominates other factors of performance.

1 Introduction

We extend a popular model, latent Dirichlet allocation (LDA), to unbounded streams of documents. In order for inference to be practical in this setting it must use constant space asymptotically and run in pseudo-linear time, perhaps $O(n)$ or $O(n \log n)$.

Canini et al. (2009) presented a method for LDA inference based on particle filters, where a sample set of models is updated online with each new token observed from a stream. In general, these models should be regularly resampled and rejuvenated using Markov Chain Monte Carlo (MCMC) steps over the history in order to improve the efficiency of the particle filter (Gilks and Berzuini, 2001). The particle filter of Canini et al. (2009) rejuvenates over independent draws from the history by storing all past observations and states. This algorithm thus has linear storage complexity and is not an online learning algorithm in a strict sense (Börschinger and Johnson, 2012).

In the current work we propose using reservoir sampling in the rejuvenation step to reduce the storage complexity of the particle filter to $O(1)$. This improvement is practically useful in the large-data setting and is also scientifically interesting in that it recovers some of the cognitive plausibility which originally motivated Börschinger and Johnson (2012). However, in experiments on the dataset studied by Canini et al. (2009), we show that rejuvenation does not benefit the particle filter's performance. Rather, performance is dominated by the effects of random initialization (a problem for which we provide a correction while abiding by the same constraints as Canini et al. (2009)). This result re-opens the question of whether rejuvenation is of practical importance in online learning for static Bayesian models.

2 Latent Dirichlet Allocation

For a sequence of N words collected into documents of varying length, we denote the j-th word as w_j, and the document it occurs in as d_i. LDA (Blei et al., 2003) "explains" the occurrence of each word by postulating that a document was generated by repeatedly: (1) sampling a topic z from $\theta^{(d)}$, the document-specific mixture of T topics, and (2) sampling a word w from $\phi^{(z)}$, the probability distribution the z-th topic defines over the vocabulary.

The goal is to infer θ and ϕ, under the model:

$$\begin{aligned} w_i \mid z_i, \phi^{(z_i)} &\sim \text{Categorical}(\phi^{(z_i)}) \\ \phi^{(z)} &\sim \text{Dirichlet}(\beta) \\ z_i \mid \theta^{(d_i)} &\sim \text{Categorical}(\theta^{(d_i)}) \\ \theta^{(d)} &\sim \text{Dirichlet}(\alpha) \end{aligned}$$

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 446–451,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

```
initialize weights $\omega_0^{(p)} = 1/P$ for $p = 1, \dots, P$
for $i = 1, \dots, N$ do
    for $p = 1, \dots, P$ do
        set $\omega_i^{(p)} = \omega_{i-1}^{(p)} \mathbf{P}(w_i \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1})$
        sample $z_i^{(p)}$ w.p. $\mathbf{P}(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i)$.
    if $\|\omega\|_2^{-2} \leq$ ESS then
        for $j \in \mathcal{R}(i)$ do
            for $p = 1, \dots, P$ do
                sample $z_j^{(p)}$ w.p.
                $\mathbf{P}(z_j^{(p)} \mid \mathbf{z}_{i \setminus j}^{(p)}, \mathbf{w}_i)$
        set $\omega_i^{(p)} = 1/P$ for each particle
```

Algorithm 1: Particle filtering for LDA.

Computing ϕ and θ exactly is generally intractable, motivating methods for approximate inference such as variational Bayesian inference (Blei et al., 2003), expectation propagation (Minka and Lafferty, 2002), and collapsed Gibbs sampling (Griffiths and Steyvers, 2004).

A limitation of these techniques is they require multiple passes over the data to obtain good samples of ϕ and θ. This requirement makes them impractical when the corpus is too large to fit directly into memory and in particular when the corpus grows without bound. This motivates online learning techniques, including sampling-based methods (Banerjee and Basu, 2007; Canini et al., 2009) and stochastic variational inference (Hoffman et al., 2010; Mimno et al., 2012; Hoffman et al., 2013). However, where these approaches generally assume the ability to draw independent samples from the full dataset, we consider the case when it is infeasible to access arbitrary elements from the history. The one existing algorithm that can be directly applied under this constraint, to our knowledge, is the streaming variational Bayes framework (Broderick et al., 2013) in which the posterior is recursively updated as new data arrives using a variational approximation.

3 Online LDA Using Particle Filters

Particle filters are a family of sequential Monte Carlo (SMC) sampling algorithms designed to estimate the posterior distribution of a system with dynamic state (Doucet et al., 2001). A particle filter approximates the posterior by a weighted sample of points, or particles, from the state space. The particle cloud is updated recursively for each new observation using importance sampling (an approach called *sequential importance sampling*).

Canini et al. (2009) apply this approach to LDA after analytically integrating out ϕ and θ, obtaining a Rao-Blackwellized particle filter (Doucet et al., 2000) that estimates the collapsed posterior $\mathbf{P}(\mathbf{z} \mid \mathbf{w})$. In this setting, the P particles are samples of the topic assignment vector $\mathbf{z}^{(p)}$, and they are propagated forward in state space one token at a time. In general, the larger P is, the more accurately we approximate the posterior; for small P, the approximation of the tails of the posterior will be particularly poor (Pitt and Shephard, 1999). However, a larger value of P increases the runtime and storage requirements of the algorithm.

We now describe the Rao-Blackwellized particle filter for LDA in detail (pseudocode is given in Algorithm 1). At the moment token i is observed, the particles form a discrete approximation of the posterior up to the $(i-1)$-th word:

$$\mathbf{P}(\mathbf{z}_{i-1} \mid \mathbf{w}_{i-1}) \approx \sum_p \omega_{i-1}^{(p)} I_{\mathbf{z}_{i-1}}(\mathbf{z}_{i-1}^{(p)})$$

where $I_{\mathbf{z}}(\mathbf{z}')$ is the indicator function, evaluating to 1 if $\mathbf{z} = \mathbf{z}'$ and 0 otherwise. Now each particle p is propagated forward by drawing a topic $z_i^{(p)}$ from the conditional posterior distribution $\mathbf{P}(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i)$ and scaling the particle weight by $\mathbf{P}(w_i \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1})$. The particle cloud now approximates the posterior up to the i-th word:

$$\mathbf{P}(\mathbf{z}_i \mid \mathbf{w}_i) \approx \sum_p \omega_i^{(p)} I_{\mathbf{z}_i}(\mathbf{z}_i^{(p)}).$$

Dropping the superscript (p) for notational convenience, the conditional posterior used in the propagation step is given by

$$\begin{aligned}\mathbf{P}(z_i|\mathbf{z}_{i-1}, \mathbf{w}_i) &\propto \mathbf{P}(z_i, w_i \mid \mathbf{z}_{i-1}, \mathbf{w}_{i-1}) \\ &= \frac{n_{z_i,i\setminus i}^{(w_i)} + \beta}{n_{z_i,i\setminus i}^{(\cdot)} + W\beta} \frac{n_{z_i,i\setminus i}^{(d_i)} + \alpha}{n_{\cdot,i\setminus i}^{(d_i)} + T\alpha}\end{aligned}$$

where $n_{z_i,i\setminus i}^{(w_i)}$ is the number of times word w_i has been assigned topic z_i so far, $n_{z_i,i\setminus i}^{(\cdot)}$ is the number of times any word has been assigned topic z_i, $n_{z_i,i\setminus i}^{(d_i)}$ is the number of times topic z_i has been assigned to any word in document d_i, and $n_{\cdot,i\setminus i}^{(d_i)}$ is the number of words observed in document d_i. The particle weights are scaled as

$$\begin{aligned}\frac{\omega_i^{(p)}}{\omega_{i-1}^{(p)}} &\propto \frac{\mathbf{P}(w_i \mid \mathbf{z}_i^{(p)}, \mathbf{w}_i)\mathbf{P}(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)})}{Q(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i)} \\ &= \mathbf{P}(w_i \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_{i-1})\end{aligned}$$

where Q is the proposal distribution for the particle state transition; in our case,

$$Q(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i) = \mathbf{P}(z_i^{(p)} \mid \mathbf{z}_{i-1}^{(p)}, \mathbf{w}_i),$$

minimizing the variance of the importance weights conditioned on $\mathbf{w}_i$ and $\mathbf{z}_{i-1}$ (Doucet et al., 2000).

Over time the particle weights tend to diverge. To combat this inefficiency, after every state transition we estimate the effective sample size (ESS) of the particle weights as $\|\omega_i\|_2^{-2}$ (Liu and Chen, 1998) and resample the particles when that estimate drops below a prespecified threshold. Several resampling strategies have been proposed (Doucet et al., 2000); we perform multinomial resampling as in Pitt and Shephard (1999) and Ahmed et al. (2011), treating the weights as unnormalized probability masses on the particles.

After resampling we are likely to have several copies of the same particle, yielding a degenerate approximation to the posterior. To reintroduce diversity to the particle cloud we take MCMC steps over a sequence of states from the history (Doucet et al., 2000; Gilks and Berzuini, 2001). We call the indices of these states the rejuvenation sequence, denoted $\mathcal{R}(i)$ (Canini et al., 2009). The transition probability for a state $j \in \mathcal{R}(i)$ is given by

$$\mathbf{P}(z_j \mid \mathbf{z}_{N\backslash j}, \mathbf{w}_N) \propto \frac{n_{z_j,N\backslash j}^{(w_j)} + \beta}{n_{z_j,N\backslash j}^{(\cdot)} + W\beta} \frac{n_{z_j,N\backslash j}^{(d_j)} + \alpha}{n_{\cdot,N\backslash j}^{(d_j)} + T\alpha}$$

where subscript $N\backslash j$ denotes counts up to token N, excluding those for token j.

The rejuvenation sequence can be chosen by the practitioner. Choosing a long sequence (large $|\mathcal{R}(i)|$) may result in a more accurate posterior approximation but also increases runtime and storage requirements. The tokens in $\mathcal{R}(i)$ may be chosen uniformly at random from the history or under a biased scheme that favors recent observations. The particle filter studied empirically by Canini et al. (2009) stored the entire history, incurring linear storage complexity in the size of the stream. Ahmed et al. (2011) instead sampled ten documents from the most recent 1000, achieving constant storage complexity at the cost of a recency bias. If we want to fit a model to a long non-i.i.d. stream, we require an unbiased rejuvenation sequence as well as sub-linear storage complexity.

4 Reservoir Sampling

Reservoir sampling is a widely-used family of algorithms for choosing an array ("reservoir") of k items. The most common example, presented in Vitter (1985) as Algorithm R, chooses k elements of a stream such that each possible subset of k elements is equiprobable. This effects sampling k items uniformly without replacement, using runtime $O(n)$ (constant per update) and storage $O(k)$.

```
Initialize $k$-element array $R$ ;
Stream $S$ ;
for $i = 1, \ldots, k$ do
    $R[i] \leftarrow S[i]$ ;
for $i = k+1, \ldots, length(S)$ do
    $j \leftarrow random(1, i)$;
    if $j \leq k$ then
        $R[j] \leftarrow S[i]$ ;
```

Algorithm 2: Algorithm R for reservoir sampling

To ensure constant space over an unbounded stream, we draw the rejuvenation sequence $\mathcal{R}(i)$ uniformly from a reservoir. As each token of the training data is ingested by the particle filter, we decide to insert that token into the reservoir, or not, independent of the other tokens in the current document. Thus, at the end of step i of the particle filter, each of the i tokens seen so far in the training sequence has an equal probability of being in the reservoir, hence being selected for rejuvenation.

5 Experiments

We evaluate our particle filter on three datasets studied in Canini et al. (2009): `diff3`, `rel3`, and `sim3`. Each of these datasets is a collection of posts under three categories from the 20 Newsgroups dataset.[1] We use a 60% training/40% testing split of this data that is available online.[2]

We preprocess the data by splitting each line on non-alphabet characters, converting the resulting tokens to lower-case, and filtering out any tokens that appear in a list of common English stop words. In addition, we remove the header of every file and filter every line that does not contain a non-trailing space (which removes embedded ASCII-encoded attachments). Finally, we shuffle the order of the documents. After these steps, we compute the vocabulary for each dataset as the set of all non-singleton types in the training data augmented with a special out-of-vocabulary symbol.

[1] `diff3`: {`rec.sport.baseball, sci.space, alt.atheism`}; `rel3`: `talk.politics.`{`misc, guns, mideast`}; and `sim3`: `comp.`{`graphics, os.ms-windows.misc, windows.x`}.

[2] `http://qwone.com/~jason/20Newsgroups/20news-bydate.tar.gz`

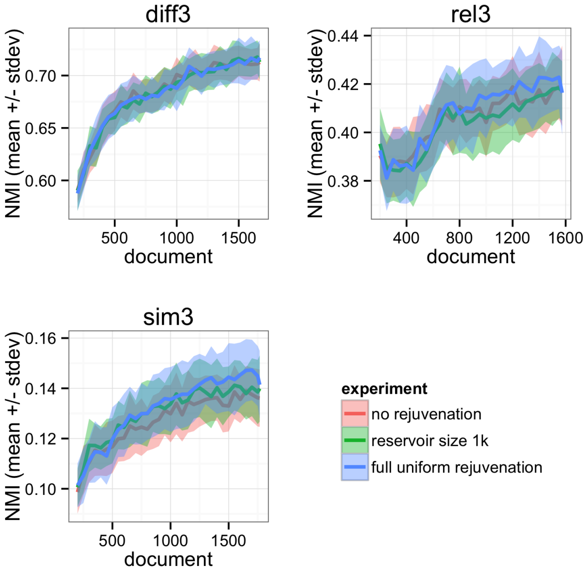

Figure 1: Fixed initialization with different reservoir sizes.

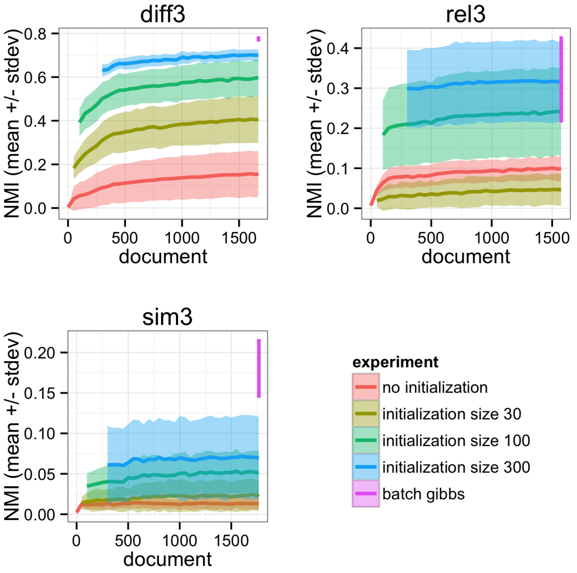

Figure 2: Variable initialization with different initialization sample sizes.

During training we report the out-of-sample NMI, calculated by holding the word proportions ϕ fixed, running five sweeps of collapsed Gibbs sampling on the test set, and computing the topic for each document as the topic assigned to the most tokens in that document. Two Gibbs sweeps have been shown to yield good performance in practice (Yao et al., 2009); we increase the number of sweeps to five after inspecting the stability on our dataset. The variance of the particle filter is often large, so for each experiment we perform 30 runs and plot the mean NMI inside bands spanning one sample standard deviation in either direction.

Fixed Initialization. Our first set of experiments has a similar parameterization[3] to the experiments of Canini et al. (2009) except we draw the rejuvenation sequence from a reservoir. We initialize the particle filter with 200 Gibbs sweeps on the first 10% of each dataset. Then, for each dataset, for rejuvenation disabled, rejuvenation based on a reservoir of size 1000, and rejuvenation based on the entire history (in turn), we perform 30 runs of the particle filter from that fixed initial model. Our results (Figure 1) resemble those of Canini et al. (2009); we believe the discrepancies are mostly attributable to differences in preprocessing.

In these experiments, the initial model was not chosen arbitrarily. Rather, an initial model that yielded out-of-sample NMI close to the initial out-of-sample NMI scores reported in the previous study was chosen from a set of 100 candidates.

Variable Initialization. We now investigate the significance of the initial model selection step used in the previous experiments. We run a new set of experiments in which the reservoir size is held fixed at 1000 and the size of the initialization sample is varied. Specifically, we vary the size of the initialization sample, in documents, between zero (corresponding to no Gibbs initialization), 30, 100, and 300, and also perform a run of batch Gibbs sampling (with no particle filter). In each case, 2000 Gibbs sweeps are performed. In these experiments, the initial models are not held fixed; for each of the 30 runs for each dataset, the initial model was generated by a different Gibbs chain. The results for these experiments, depicted in Figure 2, indicate that the size of the initialization sample improves mean NMI and reduces variance, and that the variance of the particle filter itself is dominated by the variance of the initial model.

Tuned Initialization. We observed previously that variance in the Gibbs initialization of the model contributes significantly to variance of the overall algorithm, as measured by NMI. With this in mind, we consider whether we can reduce variance in the initialization by tuning the initial model. Thus we perform a set of experiments in which we perform Gibbs initialization 20 times on the initialization set, setting the particle filter's initial model to the model out of these 20 with

[3] $T = 3, \alpha = \beta = 0.1, P = 100, ess = 20, |\mathcal{R}(i)| = 30$

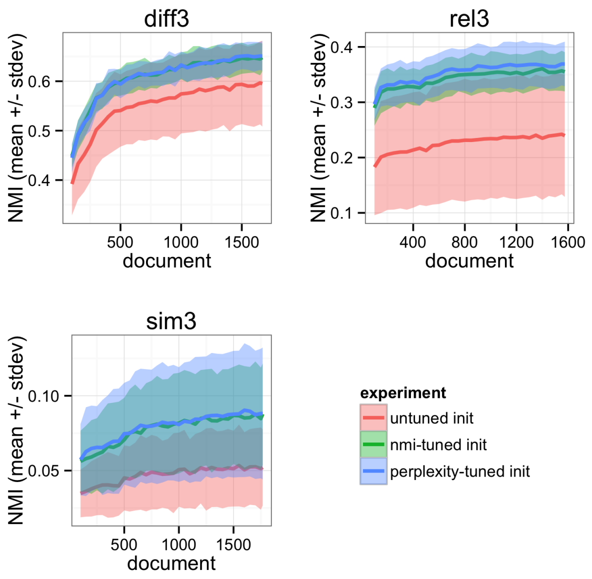

Figure 3: Variable initialization with tuning.

the highest in-sample NMI. This procedure is performed independently for each run of the particle filter. We may not always have labeled data for initialization, so we also consider a variation in which Gibbs initialization is performed 20 times on the first 80% of the initialization sample, held-out perplexity (per word) is estimated on the remaining 20%, using a first-moment particle learning approximation (Scott and Baldridge, 2013), and the particle filter is started from the model out of these 20 with the lowest held-out perplexity. The results, shown in Figure 3, show that we can ameliorate the variance due to initialization by tuning the initial model to NMI or perplexity.

6 Discussion

Motivated by a desire for cognitive plausibility, Börschinger and Johnson (2011) used a particle filter to learn Bayesian word segmentation models, following the work of Canini et al. (2009). They later showed that rejuvenation improved performance (Börschinger and Johnson, 2012), but this impaired cognitive plausibility by necessitating storage of all previous states and observations. We attempted to correct this by drawing the rejuvenation sequence from a reservoir, but our results indicate that the particle filter for LDA on our dataset is highly sensitive to initialization and not influenced by rejuvenation.

In the experiments of Börschinger and Johnson (2012), the particle cloud appears to be resampled once per utterance with a large rejuvenation sequence;[4] each particle takes many more rejuvenation MCMC steps than new state transitions and thus resembles a batch MCMC sampler. In our experiments resampling is done on the order of once per document, leading to less than one rejuvenation step per transition. Future work should carefully note this ratio: sampling history much more often than new states improves performance but contradicts the intuition behind particle filters.

We have also shown that tuning the initial model using in-sample NMI or held-out perplexity can improve mean NMI and reduce variance. Perplexity (or likelihood) is often used to estimate model performance in LDA (Blei et al., 2003; Griffiths and Steyvers, 2004; Wallach et al., 2009; Hoffman et al., 2010), and does not compare the inferred model against gold-standard labels, yet it appears to be a good proxy for NMI in our experiment. Thus, if initialization continues to be crucial to performance, at least we may have the flexibility of initializing without gold-standard labels.

We have focused on NMI as our evaluation metric for comparison with Canini et al. (2009). However, evaluation of topic models is a subject of considerable debate (Wallach et al., 2009; Yao et al., 2009; Newman et al., 2010; Mimno et al., 2011) and it may be informative to investigate the effects of initialization and rejuvenation using other metrics such as perplexity or semantic coherence.

7 Conclusion

We have proposed reservoir sampling for reducing the storage complexity of a particle filter from linear to constant. This work was motivated as an expected improvement on the model of Canini et al. (2009). However, in the process of establishing an empirical baseline we discovered that rejuvenation does not play a significant role in the experiments of Canini et al. (2009). Moreover, we found that performance of the particle filter was strongly affected by the random initialization of the model, and suggested a simple approach to reduce the variability therein without using additional data. In conclusion, it is now an open question whether—and if so, under what assumptions—rejuvenation benefits particle filters for LDA and similar static Bayesian models.

Acknowledgments We thank Frank Ferraro, Keith Levin, and Mark Dredze for discussions.

[4] The ESS threshold is P; the rejuvenation sequence is 100 or 1600 utterances, almost one sixth of the training data.

References

Amr Ahmed, Qirong Ho, Jacob Eisenstein, Eric P. Xing, Alexander J. Smola, and Choon Hui Teo. 2011. Unified analysis of streaming news. In *Proceedings of the 20th International World Wide Web Conference (WWW)*, pages 267–276.

Arindam Banerjee and Sugato Basu. 2007. Topic models over text streams: A study of batch and online unsupervised learning. In *Proceedings of the 7th SIAM International Conference on Data Mining (SDM)*, pages 431–436.

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

Benjamin Börschinger and Mark Johnson. 2011. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop (ALTA)*, pages 10–18.

Benjamin Börschinger and Mark Johnson. 2012. Using rejuvenation to improve particle filtering for Bayesian word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 85–89.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. 2013. Streaming variational Bayes. In *Advances in Neural Information Processing Systems 26 (NIPS)*.

Kevin R. Canini, Lei Shi, and Thomas L. Griffiths. 2009. Online inference of topics with latent Dirichlet allocation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 176–183.

Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. 2001. *Sequential Monte Carlo Methods in Practice*. Springer, New York.

Walter R. Gilks and Carlo Berzuini. 2001. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society*, 63(1):127–146.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, Apr.

Matthew D. Hoffman, David M. Blei, and Francis Bach. 2010. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 23 (NIPS)*.

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

Jun S. Liu and Rong Chen. 1998. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, Sep.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 262–272.

David Mimno, Matthew D. Hoffman, and David M. Blei. 2012. Sparse stochastic inference for latent Dirichlet allocation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

Thomas Minka and John Lafferty. 2002. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 352–359.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, pages 100–108.

Michael K. Pitt and Neil Shephard. 1999. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, Jun.

James G. Scott and Jason Baldridge. 2013. A recursive estimate for the predictive likelihood in a topic model. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, Mar.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation methods for topic models. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946.