SpectralLeader: Online Spectral Learning for Single Topic Models

Tong Yu[1], Branislav Kveton[2⋆], Zheng Wen[3],
Hung Bui[4], and Ole J. Mengshoel[1]

[1] Carnegie Mellon University
[2] Google Research
[3] Adobe Research
[4] Google DeepMind
tongy1@andrew.cmu.edu, bkveton@google.com, zwen@adobe.com,
bui.h.hung@gmail.com, ole.mengshoel@sv.cmu.edu

Abstract. We study the problem of learning a latent variable model online from a stream of data. Latent variable models are popular because they can explain observed data through unobserved concepts. These models have traditionally been studied in the offline setting. In the online setting, online expectation maximization (EM) is arguably the most popular approach for learning latent variable models. Although online EM is computationally efficient, it typically converges to a local optimum. In this work, we develop a new online learning algorithm for latent variable models, which we call SpectralLeader. SpectralLeader converges to the global optimum, and we derive a sublinear upper bound on its n-step regret in a single topic model. In both synthetic and real-world experiments, we show that SpectralLeader performs similarly to or better than online EM with tuned hyper-parameters.

1 Introduction

Latent variable models explain observed data through unobserved concepts. They have been successfully applied in a wide variety of fields, such as speech recognition, natural language processing, and computer vision [16, 20, 15, 5]. Despite their successes, latent variable models are typically studied in the offline setting. However, in many practical problems, a learning agent needs to learn a latent variable model online. With online algorithms, we can update the model efficiently and do not need to store all the past data. For instance, a recommender system may want to learn to cluster its users online based on their real-time behavior. This paper aims to develop algorithms for such online learning problems.

Several existing algorithms learn latent variable models online by extending expectation maximization (EM) algorithm. Those algorithms are known as online EM, and include stepwise EM [6, 13] and incremental EM [14]. Similar to offline EM, each iteration of online EM includes an E-step to fill in the values of

⋆ This work was done while the author was at Adobe Research.

latent variables based on their estimated distribution, and an M-step to update the model parameters. The main difference is that each step of online EM only uses data received recently, rather than the whole dataset. This ensures that online EM is computationally efficient and can be used to learn latent variable models online. However, similar to offline EM, online EM algorithms have one major drawback: they may converge to a local optimum and hence suffer from a non-diminishing performance loss.

To overcome these limitations, we develop an online learning algorithm that performs almost as well as the globally optimal latent variable model, which we call `SpectralLeader`. Specifically, we propose an online learning variant of the spectral method [3], which can learn the parameters of latent variable models offline with guarantees of convergence to a global optimum. Our online learning setting is defined as follows. We have a sequence of n topic models, one at each time $t \in [n]$. The prior distribution of topics can change arbitrarily over time, while the conditional distribution of words is stationary. At time t, the learning agent observes a document of words, which is sampled i.i.d. from the model at time t. The goal of the agent is to predict a sequence of model parameters with low cumulative regret with respect to the best solution in hindsight, which is constructed based on the sampling distribution of the words over n steps.

This paper makes several contributions. First, it is the first paper to formulate online learning with the spectral method as a regret minimization problem. Second, we propose `SpectralLeader`, an online learning variant of the spectral method for single topic models [3]. To reduce computational and space complexities of `SpectralLeader`, we introduce reservoir sampling. Third, we prove a sublinear upper bound on the n-step regret of `SpectralLeader`. Finally, we compare `SpectralLeader` to stepwise EM in extensive synthetic and real-world experiments. We observe that stepwise EM is sensitive to the setting of its hyper-parameters. In all experiments, `SpectralLeader` performs similarly to or better than stepwise EM with optimized hyper-parameters.

2 Related Work

The spectral method by tensor decomposition has been widely applied in different latent variable models, such as mixtures of tree graphical models [3], mixtures of linear regressions [7], hidden Markov models (HMM) [4], latent Dirichlet allocation (LDA) [2], Indian buffet process [18], and hierarchical Dirichlet process [19]. The spectral method first empirically estimates low-order moments of observations and then applies decomposition methods with a unique solution to recover the model parameters. One major advantage of the spectral method is that it learns globally optimal solutions [3].

Traditional online learning methods for latent variable models usually extend traditional iterative methods for learning latent variable models in the offline setting [6, 14, 13, 11]. Offline EM calculates sufficient statistics based on all the data, while in online EM the sufficient statistics are updated with recent data in each iteration [6, 14, 13]. Online algorithms are used to learn LDA on streaming

data [11, 1]. These online algorithms either have no convergence analysis or converge to local minima, while we aim to develop an algorithm with a theoretical guarantee of convergence to a global optimum.

An online spectral learning method has also been developed [12], with a focus on improving computational efficiency, by conducting optimization of multilinear operations in SGD and avoiding directly forming tensors. Online stochastic gradient for tensor decomposition has been analyzed [9] in a different online setting: they do not look at the online problem as regret minimization and the analysis focuses on convergence to a local minimum. In contrast, we develop an online spectral method with a theoretical guarantee of convergence to a global optimum. Further, our method is robust in the non-stochastic setting where the topics of documents are correlated over time. This non-stochastic setting has not been previously studied in the context of online spectral learning [12].

3 Spectral Method for Single Topic Models

This section introduces the offline spectral method in latent variable models. Specifically, we describe how the method works in the single topic model [3].

In the single topic model, the goal is to learn the latent topics of documents from the observed words in each document. Without loss of generality, we describe the spectral method and `SpectralLeader` (Section 5) in the setting where each document contains three words. The extension to more than three words is straightforward (Section 7). Let the number of distinct topics be K and the size of the vocabulary be d. Then our model can be viewed as a mixture model, where the three observed words $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$ are conditionally i.i.d. given topic C, which is drawn from some distribution over topics. Later in Section 4, we study a more general setting where the distribution of topic can change over time. Each word is one-hot encoded, that is $\mathbf{x}^{(l)} = e_i$ if and only if $\mathbf{x}^{(l)}$ represents word i, where $e_1, \ldots, e_d$ is the standard coordinate basis in $\mathbb{R}^d$. Define $[n] = \{1, \ldots, n\}$. The model is parameterized by the probability of each topic j, $\omega_j = P(C = j)$ for $j \in [K]$, and the conditional probability of all words $u_j \in [0, 1]^d$ given topic j. The ith entry of u_j is $u_j(i) = P(\mathbf{x}^{(l)} = e_i | C = j)$ for $i \in [d]$. To recover the model parameters, it suffices to construct a third order tensor $\bar{M}_3$ as

$$\mathbb{E}[\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \mathbf{x}^{(3)}] = \sum_{1 \le i,j,k \le d} P(\mathbf{x}^{(1)} = e_i, \mathbf{x}^{(2)} = e_j, \mathbf{x}^{(3)} = e_k)\ e_i \otimes e_j \otimes e_k.$$

We recover the parameters of the topic model by decomposing $\bar{M}_3$ as

$$\bar{M}_3 = \sum_{i=1}^{K} \omega_i u_i \otimes u_i \otimes u_i. \tag{1}$$

Unfortunately, such a decomposition is generally NP-hard [3]. Instead, we can decompose an orthogonal decomposable tensor. One way to make $\bar{M}_3$ orthogonal decomposable is by whitening. We can define a whitening matrix as $\bar{W} =$

$UA^{-1/2}$, where $A \in \mathbb{R}^{K \times K}$ is the diagonal matrix of the positive eigenvalues of $\bar{M}_2 = \mathbb{E}[\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}] = \sum_{i=1}^{K} \omega_i u_i \otimes u_i$ and $U \in \mathbb{R}^{d \times K}$ is the matrix of K eigenvectors associated with those eigenvalues. After whitening, instead of decomposing $\bar{M}_3$, we can decompose $\bar{T} = \mathbb{E}[\bar{W}^\top \mathbf{x}^{(1)} \otimes \bar{W}^\top \mathbf{x}^{(2)} \otimes \bar{W}^\top \mathbf{x}^{(3)}]$ as $\bar{T} = \sum_{i=1}^{K} \lambda_i v_i \otimes v_i \otimes v_i$ by the *power iteration method* [3]. Finally, the model parameters are recovered as $\omega_i = \frac{1}{\lambda_i^2}$ and $u_i = \lambda_i (\bar{W}^\top)^+ v_i$, where $(\bar{W}^\top)^+$ is the pseudoinverse of $\bar{W}^\top$. In practice, only a noisy realization of $\bar{T}$ is typically available, which is constructed from empirical counts. Such tensors can be decomposed approximately and the errors of such decompositions are analyzed in Theorem 5.1 of Anandkumar *et al.* [3].

4 Online Learning for Single Topic Models

We study the following online learning problem in the single topic model discussed in Section 3. We have a sequence of n topic models, one at each time $t \in [n]$. The prior distribution of topics can change arbitrarily over time, while the conditional distribution of words is stationary. We denote by $\mathbf{x}_t = (\mathbf{x}_t^{(l)})_{l=1}^3$ a tuple of one-hot encoded words in the document at time t, which is sampled i.i.d. from the model at time t. Non-stationary distributions of topics are common in practice. For instance, in the recommender system example in Section 1, user clusters tend to be correlated over time. The clusters can be viewed as topics.

We represent the distribution of words at time t by a cube $P_t = \mathbb{E}[\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}] \in [0,1]^{d \times d \times d}$. In particular, the probability of observing the triplet of words (i, j, k) at time t is

$$P_t(i,j,k) = \sum_{c=1}^{K} P_t(c) P(\mathbf{x}_t^{(1)} = e_i | c) P(\mathbf{x}_t^{(2)} = e_j | c) P(\mathbf{x}_t^{(3)} = e_k | c)\,, \tag{2}$$

where $P_t(c)$ is the prior distribution of topics at time t. This prior distribution can change arbitrarily with t.

The learning agent predicts the distribution of words $\hat{M}_{3,t-1} \in [0,1]^{d \times d \times d}$ at time t and is evaluated by its per-step loss $\ell_t(\hat{M}_{3,t-1})$. The agent aims to minimize its cumulative loss, which measures the difference between the predicted distribution $\hat{M}_{3,t-1}$ and the observations $\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}$ over time.

But what should the loss be? In this work, we define the *loss* at time t as

$$\ell_t(M) = \|\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)} - M\|_F^2, \tag{3}$$

where $\|.\|_F$ is the *Frobenius norm*. For any tensor $M \in \mathbb{R}^{d \times d \times d}$, we define its Frobenius norm as $\|M\|_F = \sqrt{\sum_{i,j,k=1}^{d} M(i,j,k)^2}$. This choice can be justified as follows. Let

$$\bar{M}_{3,n} = \frac{1}{n} \sum_{t=1}^{n} P_t = \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}] \tag{4}$$

be the average of distributions from which $\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}$ are generated in n steps. Then

$$\bar{M}_{3,n} = \operatorname*{argmin}_{M \in [0,1]^{d \times d \times d}} \sum_{t=1}^{n} \mathbb{E}[\ell_t(M)]\,, \tag{5}$$

as shown in Lemma 1 in Section 6.4. In other words, the loss function is chosen such that a natural *best solution in hindsight*, $\bar{M}_{3,n}$ in (5), is the minimizer of the cumulative loss.

With the definition of the loss function and the best solution in hindsight, the goal of the learning agent is to minimize the regret

$$R(n) = \sum_{t=1}^{n} \mathbb{E}[\ell_t(\hat{M}_{3,t-1}) - \ell_t(\bar{M}_{3,n})]\,, \tag{6}$$

where $\ell_t(\hat{M}_{3,t-1})$ is the loss of our estimated model at time t and $\ell_t(\bar{M}_{3,n})$ is the loss of the best solution in hindsight, respectively. Minimizing the regret in the online setting guarantees that the learnt model can provide more and more accurate predictions over time.

Unlike traditional online algorithms that minimize the negative log-likelihood [13], we minimize the parameter recovery loss. In the offline setting, the spectral method minimizes the recovery loss in a wide range of models [3, 7, 17].

5 Algorithm SpectralLeader

We propose SpectralLeader, an online learning algorithm for minimizing the regret in (6). Its pseudocode is in Algorithm 1. At each time t, the input is observation $(\mathbf{x}_t^{(l)})_{l=1}^3$. We maintain reservoir samples $((\mathbf{x}_z^{(l)})_{l=1}^3)_{z \in \mathcal{S}_{t-1}}$ from the previous $t-1$ time steps, where $\mathcal{S}_{t-1}$ is the time indices of these samples.

The algorithm operates as follows. First, in line 1 we construct the second-order moment from the reservoir samples, where $\Pi_2(3)$ is the set of all 2-permutations of $\{1, 2, 3\}$. Then we estimate A_{t-1} and U_{t-1} by eigendecomposition, and construct the whitening matrix W_{t-1} in line 2. After whitening, we build the third-order tensor T_{t-1} from whitened words $((W_{t-1}^\top \mathbf{x}_z^{(l)})_{l=1}^3)_{z \in \mathcal{S}_{t-1}}$ in line 3, where $\Pi_3(3)$ is the set of all 3-permutations of $\{1, 2, 3\}$. Then in line 4 with the power iteration method [3], we decompose T_{t-1} and get its eigenvalues $(\lambda_{t-1,i})_{i=1}^K$ and eigenvectors $(v_{t-1,i})_{i=1}^K$. Finally, in line 5 we recover the parameters of the model, the probability of topics $(\omega_{t-1,i})_{i=1}^K$ and the conditional probability of words $(u_{t-1,i})_{i=1}^K$. After recovering the parameters, we update the set of reservoir samples in lines 6 to 13. We keep m_r reservoir samples $\mathbf{x}_z$, $z \in [t-1]$. When $t \leq m_r$, the new observation $(\mathbf{x}_t^{(l)})_{l=1}^3$ is added to the reservoir. When $t > m_r$, $(\mathbf{x}_t^{(l)})_{l=1}^3$ replaces a random observation in the reservoir with probability $m_r/(t-1)$.

In SpectralLeader, we use reservoir sampling for computational efficiency reasons. Without reservoir sampling, the operations in lines 1 and 3 of Algorithm 1 would depend on t because all past observations are used to construct

Algorithm 1: `SpectralLeader` at time t

Input: Observations $(\mathbf{x}_t^{(l)})_{l=1}^3$

1 $M_{2,t-1} \leftarrow \frac{1}{|\mathcal{S}_{t-1}||\Pi_2(3)|} \sum_{z \in \mathcal{S}_{t-1}} \sum_{\pi \in \Pi_2(3)} \mathbf{x}_z^{(\pi(1))} \otimes \mathbf{x}_z^{(\pi(2))}$

2 $W_{t-1} \leftarrow U_{t-1} A_{t-1}^{-1/2}$, where $A_{t-1} \in \mathbb{R}^{K \times K}$ is the diagonal matrix of K positive eigenvalues of $M_{2,t-1}$ and $U_{t-1} \in \mathbb{R}^{d \times K}$ is the matrix of eigenvectors associated with these positive eigenvalues

3 $T_{t-1} \leftarrow \frac{1}{|\mathcal{S}_{t-1}||\Pi_3(3)|} \sum_{z \in \mathcal{S}_{t-1}} \sum_{\pi \in \Pi_3(3)} W_{t-1}^\top \mathbf{x}_z^{(\pi(1))} \otimes W_{t-1}^\top \mathbf{x}_z^{(\pi(2))} \otimes W_{t-1}^\top \mathbf{x}_z^{(\pi(3))}$

4 Obtain $(\lambda_{t-1,i})_{i=1}^K$ and $(v_{t-1,i})_{i=1}^K$ from T_{t-1} by power iteration method

5 $\omega_{t-1,i} \leftarrow \frac{1}{\lambda_{t-1,i}^2}, \quad u_{t-1,i} \leftarrow \lambda_{t-1,i} (W_{t-1}^\top)^+ v_{t-1,i}$ for all $i \in [K]$

6 Generate a random number $a \in [0, 1]$

7 **if** $t \leq m_r$ **then**

8 $\quad \mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{t\}$

9 **else if** $a \leq m_r/(t-1)$ **then**

10 $\quad$ Remove a random element of $\mathcal{S}_{t-1}$

11 $\quad \mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{t\}$

12 **else**

13 $\quad \mathcal{S}_t \leftarrow \mathcal{S}_{t-1}$

Output: Model parameters $\omega_{t-1,i}$ and $u_{t-1,i}$

$M_{2,t-1}$ and T_{t-1}. Besides, the whitening operation in line 3 would depend on t because all past observations are whitened by a matrix W_{t-1} that changes with t. With reservoir sampling, we approximate $M_{2,t-1}$, T_{t-1}, and W_{t-1} with m_r reservoir samples. We discuss how to set m_r in Section 6.2.

6 Analysis

In this section, we bound the regret of `SpectralLeader`. In Section 6.1, we analyze the regret of `SpectralLeader` without reservoir sampling in the noise-free setting. In this setting, at time t the agent knows the distribution of words $(P_z)_{z=1}^{t-1}$. The regret is due to not knowing P_t at time t. In Section 6.2, we analyze the regret of `SpectralLeader` with reservoir sampling in the noise-free setting. In this setting, the agent knows $(P_z)_{z \in \mathcal{S}_{t-1}}$ at time t, which is a random sample of $(P_z)_{z=1}^{t-1}$. In comparison to Section 6.1, the additional regret is due to reservoir sampling. In Section 6.3, we discuss the regret of `SpectralLeader` with reservoir sampling in the noisy setting. In this setting, the agent approximates each distribution P_z with its single empirical observation $(\mathbf{x}_z^{(l)})_{l=1}^3$, for any $z \in \mathcal{S}_{t-1}$. In comparison to Section 6.2, the additional regret is due to noisy observations. All supplementary lemmas are stated and proved in Section 6.4.

6.1 Noise-Free Setting

We first analyze an idealized variant of SpectralLeader, where the agent knows $(P_z)_{z=1}^{t-1}$ at time t. In this setting, the algorithm is similar to Algorithm 1, except that lines 1 and 3 are replaced, respectively, by $\bar{M}_{2,t-1} = \frac{1}{t-1}\sum_{z=1}^{t-1}\mathbb{E}[\mathbf{x}_z^{(1)} \otimes \mathbf{x}_z^{(2)}]$ and $\bar{T}_{t-1} = \frac{1}{t-1}\sum_{z=1}^{t-1}\mathbb{E}[\bar{W}_{t-1}^\top \mathbf{x}_z^{(1)} \otimes \bar{W}_{t-1}^\top \mathbf{x}_z^{(2)} \otimes \bar{W}_{t-1}^\top \mathbf{x}_z^{(3)}]$.

We denote by $\bar{W}_{t-1}$ the corresponding whitening matrix in line 2, and by $\bar{\omega}_{t-1,i}$ and $\bar{u}_{t-1,i}$ the estimated model parameters. In this noise-free setting, the power iteration method in line 4 is exact. Therefore, the prediction of the learning agent at time t satisfies $\hat{M}_{3,t-1} = \sum_{i=1}^K \bar{\omega}_{t-1,i}\, \bar{u}_{t-1,i} \otimes \bar{u}_{t-1,i} \otimes \bar{u}_{t-1,i} = \bar{M}_{3,t-1}$ for any t, according to (1).

Theorem 1. *Let $\hat{M}_{3,t-1} = \bar{M}_{3,t-1}$ at all times $t \in [n]$. Then*

$$R(n) \le 4\sqrt{d^3}\log n\,.$$

Proof. From Lemma 2, $\sum_{t=1}^n \mathbb{E}[\ell_t(\bar{M}_{3,n})] \ge \sum_{t=1}^n \mathbb{E}[\ell_t(\bar{M}_{3,t})]$. Now note that $\hat{M}_{3,t-1} = \bar{M}_{3,t-1}$ at any time t, and therefore

$$R(n) = \sum_{t=1}^n \mathbb{E}[\ell_t(\bar{M}_{3,t-1}) - \ell_t(\bar{M}_{3,n})] \le \sum_{t=1}^n \mathbb{E}[\ell_t(\bar{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t})]\,.$$

At any time t and for any $\mathbf{x}_t$,

$$\begin{aligned}
\ell_t(\bar{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t}) &\le 4\|\bar{M}_{3,t-1} - \bar{M}_{3,t}\|_F \\
&= 4\left\|\frac{1}{t-1}\sum_{t'=1}^{t-1} P_{t'} - \frac{1}{t}\sum_{t'=1}^{t} P_{t'}\right\|_F \\
&= \frac{4}{t}\left\|\frac{1}{t-1}\sum_{t'=1}^{t-1} P_{t'} - P_t\right\|_F \le \frac{4\sqrt{d^3}}{t}\,,
\end{aligned}$$

where the first inequality is by Lemma 3 and the second inequality is from the fact that all entries of P_t are in $[0,1]$ at any time $t \in [n]$. Therefore, $R(n) \le \sum_{t=1}^n \frac{4\sqrt{d^3}}{t} \le 4\sqrt{d^3}\log n$. This concludes our proof. ■

6.2 Reservoir Sampling in Noise-Free Setting

We further analyze SpectralLeader with reservoir sampling in the noise-free setting. As discussed in Section 5, without reservoir sampling, the construction time of the decomposed tensor at time t would grow linearly with t, which is undesirable. In this setting, the algorithm is similar to Algorithm 1, except that lines 1 and 3 are replaced, respectively, by $\tilde{M}_{2,t-1} = \frac{1}{|\mathcal{S}_{t-1}|}\mathbb{E}[\sum_{z\in\mathcal{S}_{t-1}} \mathbf{x}_z^{(1)} \otimes \mathbf{x}_z^{(2)}]$ and $\tilde{T}_{t-1} = \frac{1}{|\mathcal{S}_{t-1}|}\mathbb{E}[\sum_{z\in\mathcal{S}_{t-1}} \tilde{W}_{t-1}^\top \mathbf{x}_z^{(1)} \otimes \tilde{W}_{t-1}^\top \mathbf{x}_z^{(2)} \otimes \tilde{W}_{t-1}^\top \mathbf{x}_z^{(3)}]$, where $\mathcal{S}_{t-1}$ are indices of the reservoir samples at time t. We denote by $\tilde{W}_{t-1}$ the corresponding whitening matrix in line 2, and by $\tilde{\omega}_{t-1,i}$ and $\tilde{u}_{t-1,i}$ the estimated model

parameters. As in Section 6.1, the power iteration method in line 4 is exact, and therefore the prediction of the learning agent at time t satisfies $\hat{M}_{3,t-1} = \sum_{i=1}^{K} \tilde{\omega}_{t-1,i}\, \bar{u}_{t-1,i} \otimes \tilde{u}_{t-1,i} \otimes \tilde{u}_{t-1,i} = \tilde{M}_{3,t-1}$ for any t. The main result of this section is stated below.

Theorem 2. *Let all corresponding entries of $\tilde{M}_{3,t-1}$ and $\bar{M}_{3,t-1}$ be close with a high probability,*

$$P(\exists t, i, j, k : |\tilde{M}_{3,t-1}(i,j,k) - \bar{M}_{3,t-1}(i,j,k)| \geq \epsilon) = \delta \tag{7}$$

for some small $\epsilon \in [0,1]$ and $\delta \in [0,1]$. Let $\hat{M}_{3,t-1} = \tilde{M}_{3,t-1}$ at all times $t \in [n]$. Then

$$R(n) \leq 4\sqrt{d^3}\epsilon n + 4\sqrt{d^3}\delta n + 4\sqrt{d^3}\log n\,.$$

Proof. From the definition of $R(n)$ in (6) and the bound in Theorem 1,

$$\begin{aligned} R(n) &= \sum_{t=1}^{n} \mathbb{E}[\ell_t(\tilde{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t-1})] + \sum_{t=1}^{n} \mathbb{E}[\ell_t(\bar{M}_{3,t-1}) - \ell_t(\bar{M}_{3,n})] \\ &\leq \sum_{t=1}^{n} \mathbb{E}[\ell_t(\tilde{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t-1})] + 4\sqrt{d^3}\log n\,. \end{aligned}$$

We bound the first term above as follows. Suppose that the event in (7) does not happen. Then $\ell_t(\tilde{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t-1}) \leq 4\sqrt{d^3}\epsilon$, from Lemma 3 and the fact that all corresponding entries of $\tilde{M}_{3,t-1}$ and $\bar{M}_{3,t-1}$ are ϵ-close. Now suppose that the event in (7) happens. Then $\ell_t(\tilde{M}_{3,t-1}) - \ell_t(\bar{M}_{3,t-1}) \leq 4\sqrt{d^3}$, from Lemma 3 and the fact all entries of $\tilde{M}_{3,t-1}$ and $\bar{M}_{3,t-1}$ are in $[0,1]$. Finally, note that the event in (7) happens with probability δ. Now we chain all inequalities and obtain $R(n) \leq 4\sqrt{d^3}\epsilon n + 4\sqrt{d^3}\delta n + 4\sqrt{d^3}\log n$. ■

Note that the reservoir at time t, $\mathcal{S}_{t-1} \in [t-1]$, is a random sample of size m_r for any $t > m_r + 1$. Therefore, from Hoeffding's inequality [10] and the union bound, we get that

$$\begin{aligned} \delta &= P(\exists t, i, j, k : |\tilde{M}_{3,t-1}(i,j,k) - \bar{M}_{3,t-1}(i,j,k)| \geq \epsilon) \\ &\leq 2 \sum_{t=m_r+2}^{n} d^3 \exp[-2\epsilon^2 m_r] \leq 2d^3 n \exp[-2\epsilon^2 m_r]\,. \end{aligned}$$

In addition, let the size of the reservoir be $m_r = \epsilon^{-2}\log(d^3 n)$. Then the regret bound in Theorem 2 simplifies to $R(n) < 4\sqrt{d^3}\epsilon n + 4\sqrt{d^3}\log n + 8$. This bound can be sublinear in n only if $\epsilon = o(1)$. Moreover, the definition of m_r and $m_r \leq n$ imply that $\epsilon \geq \sqrt{\log(d^3 n)/n}$. As a result of these constraints, the range of reasonable values for ϵ is $[\sqrt{\log(d^3 n)/n}, o(1))$.

For any $\epsilon \in [\sqrt{\log(d^3 n)/n}, o(1))$, the regret $R(n)$ is sublinear in n, where ϵ is a tunable parameter. At lower values of ϵ, $R(n) = O(\sqrt{n})$ but the reservoir size approaches n. At higher values of ϵ, the reservoir size is $O(\log n)$ but $R(n)$ approaches n.

6.3 Reservoir Sampling in Noisy Setting

Finally, we discuss the regret of SpectralLeader with reservoir sampling in the noisy setting. In this setting, the analyzed algorithm is Algorithm 1. The predicted distribution at time t is $\hat{M}_{3,t-1} = \sum_{i=1}^{K} \omega_{t-1,i}\, u_{t-1,i} \otimes u_{t-1,i} \otimes u_{t-1,i}$.

From the definition of $R(n)$ and our earlier analysis, $R(n)$ can be decomposed and bounded from above as

$$R(n) \leq \sum_{t=1}^{n} \mathbb{E}[\ell_t(\hat{M}_{3,t-1}) - \ell_t(\tilde{M}_{3,t-1})] + 4\sqrt{d^3}\epsilon n + 4\sqrt{d^3}\log n + 8 \qquad (8)$$

when the size of the reservoir is $m_r = \epsilon^{-2}\log(d^3 n)$.

Suppose that $m_r \to \infty$ as $n \to \infty$, for instance by setting $\epsilon = n^{-\frac{1}{4}}$. Under this assumption, $M_{2,t-1}$ in SpectralLeader approaches $\tilde{M}_{2,t-1}$ (Section 6.2) because $M_{2,t-1}$ is an empirical estimator of $\tilde{M}_{2,t-1}$ on m_r observations. By Weyl's and Davis-Kahan theorems [21, 8], the eigenvalues and eigenvectors of $M_{2,t-1}$ approach those of $\tilde{M}_{2,t-1}$ as $m_r \to \infty$, and thus the whitening matrix W_{t-1} in SpectralLeader approaches $\tilde{W}_{t-1}$ (Section 6.2). Since T_{t-1} in SpectralLeader is an empirical estimator of $\tilde{T}_{t-1}$ (Section 6.2) on m_r whitened observations and $W_{t-1} \to \tilde{W}_{t-1}$, we have $T_{t-1} \to \tilde{T}_{t-1}$ as $m_r \to \infty$. In our online setting (Section 4), over time the data samples are generated by topic models with the same conditional distribution of words. Thus, the reservoir samples are actually generated by a topic model with this conditional distribution and an arbitrary distribution of topics. Therefore, Theorem 5.1 of Anandkumar *et al.* [3] applies: the eigenvalues and eigenvectors of T_{t-1} approach those of $\tilde{T}_{t-1}$ as $T_{t-1} \to \tilde{T}_{t-1}$. This implies that $\hat{M}_{3,t-1} \to \tilde{M}_{3,t-1}$, as all quantities that $\hat{M}_{3,t-1}$ and $\tilde{M}_{3,t-1}$ depend on approach each other as $m_r \to \infty$. Therefore, $\lim_{n\to\infty} \lim_{t\to n}(\ell_t(\hat{M}_{3,t-1}) - \ell_t(\tilde{M}_{3,t-1})) = 0$ and the regret bound in (8) is $o(n)$, sublinear in n, as $n \to \infty$.

6.4 Technical Lemmas

Lemma 1. *Let* $\ell_t(M) = \|\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)} - M\|_F^2$. *Then*

$$\bar{M}_{3,n} = \operatorname*{argmin}_{M \in [0,1]^{d\times d\times d}} \sum_{t=1}^{n} \mathbb{E}[\ell_t(M)]\,, \qquad (9)$$

where $\bar{M}_{3,n}$ *is defined in* (4).

Proof. It is sufficient to show that

$$\bar{M}_{3,n}(i,j,k) = \operatorname*{argmin}_{y\in[0,1]} \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}[(\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}(i,j,k) - y)^2] \qquad (10)$$

for any (i,j,k), where $\bar{M}_{3,n}(i,j,k)$ and $\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}(i,j,k)$ are the (i,j,k)-th entries of tensors $\bar{M}_{3,n}$ and $\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}$, respectively. To prove the claim,

let $f(y) = \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}[(\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}(i,j,k) - y)^2]$. Then

$$\frac{\partial}{\partial y} f(y) = 2y - \frac{2}{n}\sum_{t=1}^{n}\mathbb{E}[\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}(i,j,k)]\,.$$

Now we put the derivative equal to zero and get $y = \bar{M}_{3,n}(i,j,k)$. ■

Lemma 2. *For any n, $\sum_{t=1}^{n}\mathbb{E}[\ell_t(\bar{M}_{3,n})] \geq \sum_{t=1}^{n}\mathbb{E}[\ell_t(\bar{M}_{3,t})]$.*

Proof. We prove this claim by induction. First, suppose that $n = 0$. Then trivially $\mathbb{E}[\ell_t(\bar{M}_{3,0})] \geq \mathbb{E}[\ell_t(\bar{M}_{3,0})]$. Second, by induction hypothesis, we have that

$$\sum_{t=1}^{n-1}\mathbb{E}[\ell_t(\bar{M}_{3,n-1})] \geq \sum_{t=1}^{n-1}\mathbb{E}[\ell_t(\bar{M}_{3,t})]\,. \tag{11}$$

Then

$$\begin{aligned}\sum_{t=1}^{n}\mathbb{E}[\ell_t(\bar{M}_{3,n})] &= \sum_{t=1}^{n-1}\mathbb{E}[\ell_t(\bar{M}_{3,n})] + \mathbb{E}[\ell_n(\bar{M}_{3,n})] \\ &\geq \sum_{t=1}^{n-1}\mathbb{E}[\ell_t(\bar{M}_{3,n-1})] + \mathbb{E}[\ell_n(\bar{M}_{3,n})] \geq \sum_{t=1}^{n}\mathbb{E}[\ell_t(\bar{M}_{3,t})]\,,\end{aligned}$$

where the first inequality is from (9) and the second inequality is from (11). ■

Lemma 3. *For any tensors $M \in [0,1]^{d\times d\times d}$ satisfying $\sum_{i,j,k=1}^{d} M(i,j,k) = 1$, and $M' \in [0,1]^{d\times d\times d}$ satisfying $\sum_{i,j,k=1}^{d} M'(i,j,k) = 1$, we have*

$$\ell_t(M) - \ell_t(M') \leq 4\|M - M'\|_F.$$

Proof. The proof follows from elementary algebra

$$\begin{aligned}&\ell_t(M) - \ell_t(M') \\ &\quad= (\ell_t^{\frac{1}{2}}(M) + \ell_t^{\frac{1}{2}}(M'))(\ell_t^{\frac{1}{2}}(M) - \ell_t^{\frac{1}{2}}(M')) \\ &\quad\leq (\ell_t^{\frac{1}{2}}(M) + \ell_t^{\frac{1}{2}}(M'))\|M - M'\|_F \\ &\quad= \left(\|\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)} - M\|_F + \|\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)} - M'\|_F\right)\|M - M'\|_F \\ &\quad\leq \left(2\|\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}\|_F + \|M\|_F + \|M'\|_F\right)\|M - M'\|_F \\ &\quad= (2 + \|M\|_F + \|M'\|_F)\,\|M - M'\|_F \leq 4\|M - M'\|_F\,.\end{aligned}$$

The first equality is from $\alpha^2 - \beta^2 = (\alpha+\beta)(\alpha-\beta)$. The first inequality is from the reverse triangle inequality. The second inequality is from the triangle inequality. The third equality is from the fact that only one entry of $\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)} \otimes \mathbf{x}_t^{(3)}$ is 1 and all the rest are 0, by the definition of $(\mathbf{x}_t^{(l)})_{l=1}^{3}$ in Section 4. The third inequality is from $\|M\|_F = \sqrt{\sum_{i,j,k=1}^{d} M(i,j,k)^2} \leq \sqrt{\sum_{i,j,k=1}^{d} |M(i,j,k)|} = 1$, and similarly $\|M'\|_F \leq 1$, which follows from the fact that tensors M and M' represent distributions with all entries in $[0,1]$ and summing up to 1. ■

7 Extensions of SpectralLeader

SpectralLeader can be extended as follows. First, it can easily be extended to documents with $L \geq 3$ words. Then, at time t, T_{t-1} in Algorithm 1 is calculated by averaging over all $\binom{L}{3}3!$ ordered triplets of words [3, 22]. The analysis essentially remains the same and the regret bound still holds. Second, SpectralLeader can be extended to more complicated latent variable models. For example, we can use SpectralLeader in Algorithm 1 to learn Gaussian mixture models (GMM) online, by redefining $M_{2,t-1}$ and T_{t-1} according to Theorem 3.2 of Anandkumar *et al.* [3]. The current analysis does not apply to GMM, since P_t in (2) is not bounded in GMM. We leave the analysis of SpectralLeader in such more complicated models for future work.

8 Experiments

In this section, we evaluate SpectralLeader and compare it empirically with stepwise EM [6]. We experiment with both stochastic and non-stochastic synthetic problems, as well as with two real-world problems.

Our chosen baseline is stepwise EM [6], an online EM algorithm. We choose this baseline as it outperforms other online EM algorithms [13], such as incremental EM [14]. Stepwise EM has two key tuning parameters: the step-size reduction power α and the mini-batch size m [13, 6]. The smaller the α, the faster the old sufficient statistics are forgotten. The mini-batch size m is the number of documents to calculate the sufficient statistics for each update of stepwise EM. With larger m, we can usually add stability to stepwise EM. We compared SpectralLeader to stepwise EM with varying α and m.

All compared algorithms are evaluated by their models at time t, $\theta_{t-1} = ((\omega_{t-1,i})_{i=1}^K, (u_{t-1,i})_{i=1}^K)$, which are learned from the first $t-1$ steps. We report two metrics: *average negative predictive log-likelihood up to step* n, $\mathcal{L}_n^{(1)} = \frac{1}{n}\sum_{t=2}^n \left(-\log \sum_{i=1}^K P_{\theta_{t-1}}(C=i) \prod_{l=1}^L P_{\theta_{t-1}}(\mathbf{x} = \mathbf{x}_t^{(l)} \mid C = i)\right)$, where L is the number of observed words in each document; and *average recovery error up to step* n, $\mathcal{L}_n^{(2)} = \frac{1}{n}\sum_{t=2}^n \|M_{3,*} - \hat{M}_{3,t-1}\|_F^2$. The latter metric is the average difference between the distribution in hindsight $M_{3,*}$ and the predicted distribution $\hat{M}_{3,t-1}$ at time t, and measures the parameter reconstruction error. Specifically, $M_{3,*} = \sum_{i=1}^K \omega_{*,i} u_{*,i} \otimes u_{*,i} \otimes u_{*,i}$ and $\hat{M}_{3,t-1} = \sum_{i=1}^K \omega_{t-1,i} u_{t-1,i} \otimes u_{t-1,i} \otimes u_{t-1,i}$, where $\theta^* = ((\omega_{*,i})_{i=1}^K, (u_{*,i})_{i=1}^K)$ are the parameters of the unknown model. In synthetic problems, we know θ^*. In real-world problems, we learn θ^* by the spectral method since we have all data in advance. The recovery error is related to the regret, through the relation of our loss function and the Frobenius norm in Lemma 3. Note that EM in the offline setting minimizes the negative log-likelihood, while the spectral method in the offline setting minimizes the recovery error of tensors. All reported results are averaged over 10 runs.

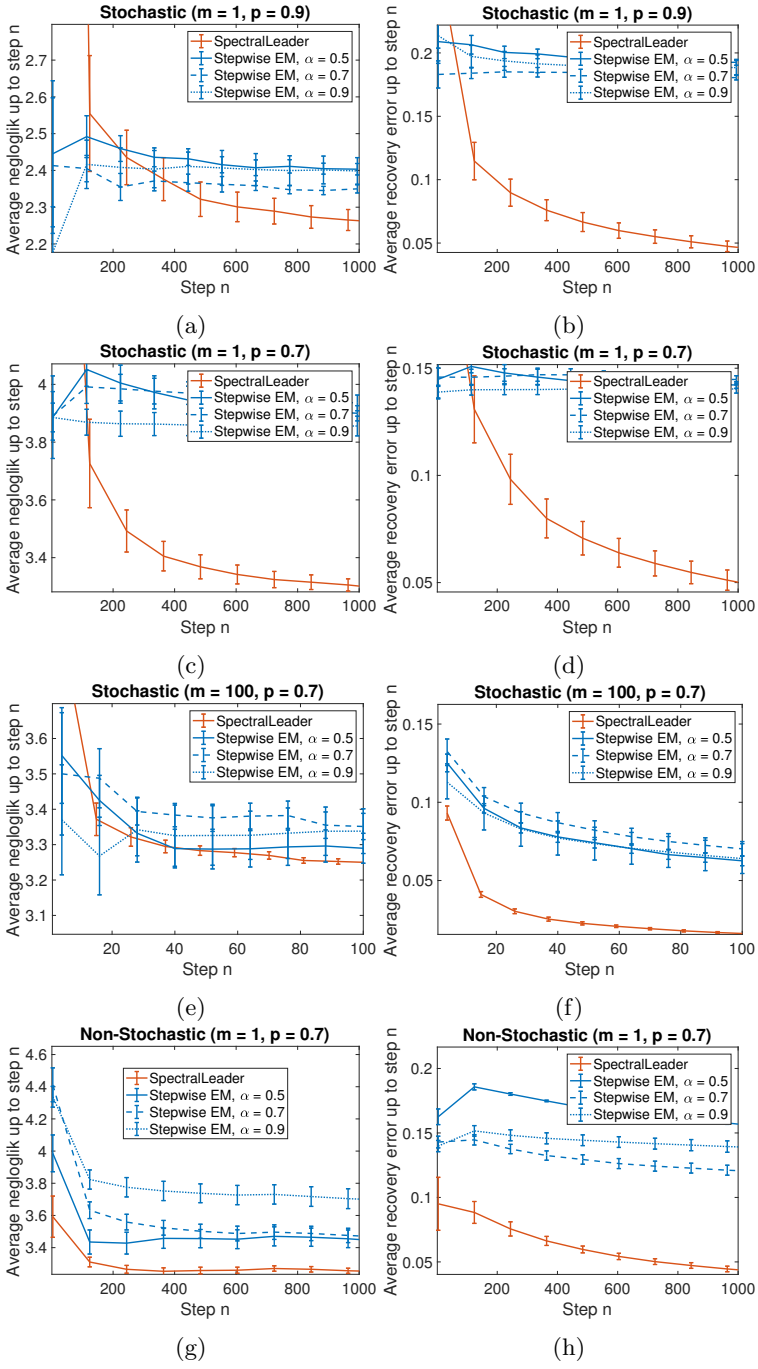

Fig. 1: Evaluation on stochastic synthetic problems and non-stochastic synthetic problems. We compare `SpectralLeader` to stepwise EM with varying step-size reduction power α and mini-batch size m. The first column shows results under metric $\mathcal{L}_n^{(1)}$ and the second column shows results under metric $\mathcal{L}_n^{(2)}$.

8.1 Synthetic Problems

Stochastic Synthetic Problems In this stochastic setting, the topic of the document at all times t is sampled i.i.d. from a fixed distribution. This setting represents a scenario where the sequence of topics is not correlated. The number of distinct topics is $K = 3$, the vocabulary size is $d = 3$, and each document has 3 observed words. In practice, some topics are typically more popular than others. Therefore, we sample topics as follows. At each time, the topic C is randomly sampled from the distribution where $P(C = 1) = 0.15$, $P(C = 2) = 0.35$, and $P(C = 3) = 0.5$. Given the topic, the conditional probability of words is $P(\mathbf{x} = e_i|C = j) = p$ when $i = j$, and $P(\mathbf{x} = e_i|C = j) = \frac{1-p}{2}$ when $i \neq j$. With smaller p, the conditional distribution of words given different topic becomes similar, and the difficulty of distinguishing different topics increases. For $m = 1$, we evaluate on two problems where $p = 0.7$ and $p = 0.9$. For $m = 100$, we further focus on the more difficult problem where $p = 0.7$. We show the results before the different methods converge: for $m = 1$, we report results before $n = 1000$, and for $m = 100$ we report both results before $n = 100$.

Results for the stochastic setting are reported in Figure 1. We observe three trends. First, under metric $\mathcal{L}_n^{(1)}$, stepwise EM is very sensitive to its parameters α and m, while `SpectralLeader` is competitive or even better, compared to stepwise EM with its best α and m. For example, the best α is 0.7 in Figure 1a, and the best α is 0.9 in Figure 1c. Even for the same problem with different m, the best α is different: the best α is 0.9 in Figure 1c, while the best α is 0.5 in Figure 1e. In all cases, `SpectralLeader` performs the best. Second, similar to [13], stepwise EM improves when the mini-batch size increases to $m = 100$. But `SpectralLeader` still performs better compared to stepwise EM with its best α. Third, `SpectralLeader` performs much better than stepwise EM under metric $\mathcal{L}_n^{(2)}$. These results indicate that a careful grid search of α and m is usually needed to optimize stepwise EM. Such grid search in the online setting is nearly impossible, since future data are unknown in advance. In contrast, `SpectralLeader` is very competitive without any parameter tuning.

Non-Stochastic Synthetic Problems The non-stochastic setting is the same as the stochastic setting, except that topics of the documents are strongly correlated over time. We look at an extreme case of correlated topics in the steaming data. In each batch of 100 steps, sequentially we have 15 documents from topic 1, 35 documents from topic 2, and 50 documents from topic 3. We focus on the more difficult problem where $p = 0.7$.

Our results in this non-stochastic setting are reported in Figures 1g and 1h. For stepwise EM, the α leading to lowest negative log-likelihood is 0.5. This result matches well the fact that the smaller the α, the faster the old sufficient statistics are forgotten, and the faster stepwise EM adapts to the non-stochastic setting. `SpectralLeader` is even better than stepwise EM with $\alpha = 0.5$. Note that $\alpha = 0.5$ is the smallest valid value of α for stepwise EM [13].

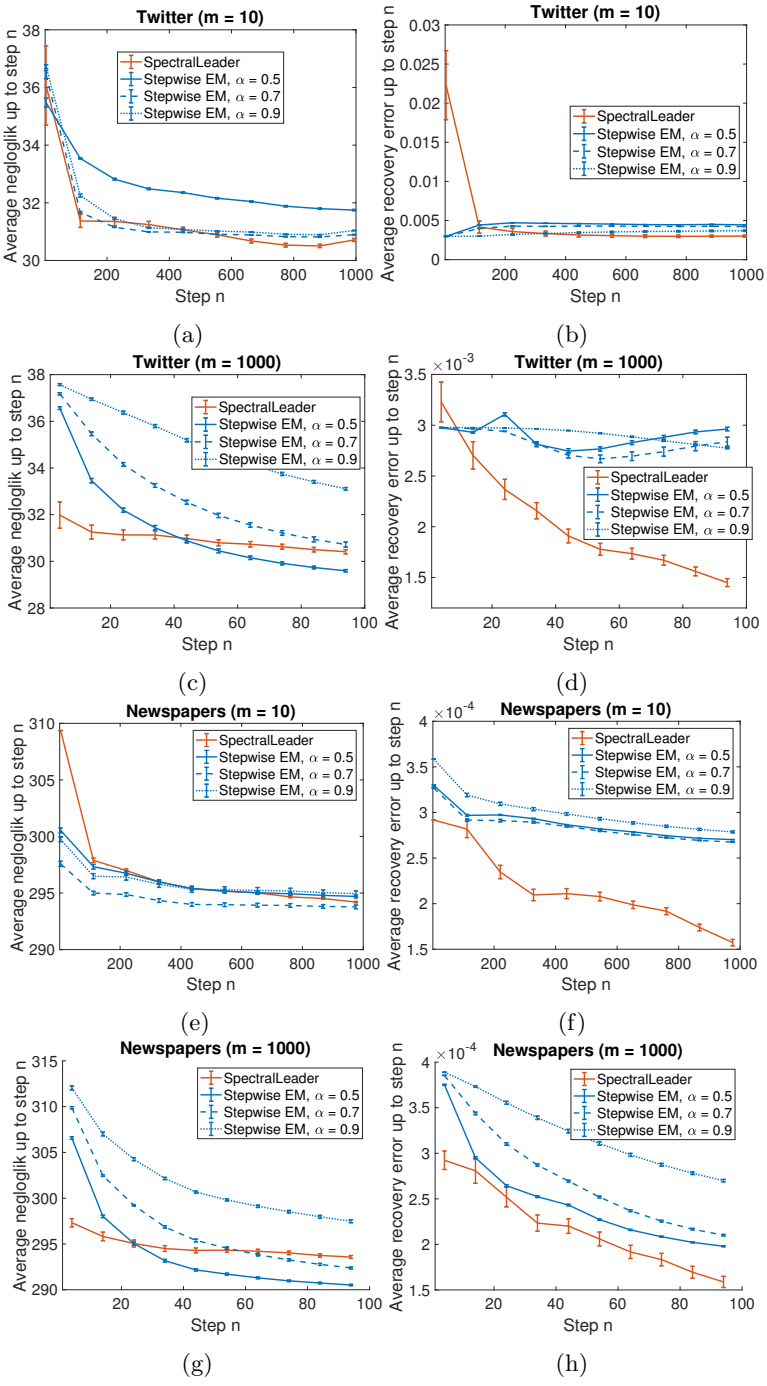

Fig. 2: Evaluation on real-world datasets. We compare `SpectralLeader` to stepwise EM with varying step-size reduction power α and mini-batch size m. The first column shows results under metric $\mathcal{L}_n^{(1)}$ and the second column shows results under metric $\mathcal{L}_n^{(2)}$.

8.2 Real World Problems

In this section, we evaluate on Newspapers data[5] over multiple years and Twitter data[6] during the 2016 United States elections. They provide streaming data with timestamps and the distributions of topics change over time. After preprocessing, we retain the 500 most frequent words in the vocabulary. We set $K = 5$. We evaluate all algorithms on 100K documents.[7] We compare SpectralLeader to stepwise EM with multiple α, and mini-batch sizes $m = 10$ and $m = 1000$. We show the results before the different methods converge: for $m = 10$, we report results before $n = 1000$, and for $m = 1000$ we report results before $n = 100$. To handle large-scale streaming data, such as 5M words in Newspapers data, we introduce reservoir sampling, and set the window size of reservoir to 10,000.

Our results are reported in Figure 2. We observe four major trends. First, under metric $\mathcal{L}_n^{(2)}$, SpectralLeader performs better than stepwise EM. Second, under metric $\mathcal{L}_n^{(1)}$, for $m = 10$ versus $m = 1000$, the optimal α for stepwise EM are different on both datasets. Third, when $m = 10$, under $\mathcal{L}_n^{(1)}$, SpectralLeader performs competitive with or better than stepwise EM with its best α. Fourth, when $m = 1000$, under $\mathcal{L}_n^{(1)}$, SpectralLeader is not as good as stepwise EM with its best α. However, directly using SpectralLeader without tuning any parameters can still provide good performance. These results suggest that, even when the mini-batch size is large, SpectralLeader is still very useful under the log-likelihood metric. In practice, we can quickly achieve reasonable results with SpectralLeader without any parameter tuning.

9 Conclusions

We develop SpectralLeader, a novel online learning algorithm for latent variable models. In an instance of a single topic model, we define a novel per-step loss function, prove that SpectralLeader converges to a global optimum, and derive a sublinear regret bound for SpectralLeader. Our experimental results suggest that SpectralLeader performs similarly to or better than a fine-tuned online EM. In future work, we want to extend our method to more complicated latent-variable models, such as HMMs and LDA [3].

Acknowledgment

This work is supported, in part, by funding from Adobe and Intel to CMU.

[5] Please see https://www.kaggle.com/snapcrack/all-the-news.
[6] Please see https://www.kaggle.com/kinguistics/election-day-tweets.
[7] The per-step computational cost of SpectralLeader is larger than that of stepwise EM, when the number of topics, number of observed words and vocabulary size increase. We leave improving the efficiency of SpectralLeader as future work.

References

1. Amoualian, H., Clausel, M., Gaussier, E., Amini, M.R.: Streaming-lda: A copula-based approach to modeling topic dependencies in document streams. In: KDD. pp. 695–704. ACM (2016)
2. Anandkumar, A., Foster, D.P., Hsu, D.J., Kakade, S.M., Liu, Y.K.: A spectral algorithm for latent Dirichlet allocation. In: NIPS. pp. 917–925 (2012)
3. Anandkumar, A., Ge, R., Hsu, D.J., Kakade, S.M., Telgarsky, M.: Tensor decompositions for learning latent variable models. JMLR **15**(1), 2773–2832 (2014)
4. Anandkumar, A., Hsu, D., Kakade, S.M.: A method of moments for mixture models and hidden Markov models. In: COLT. pp. 33–1 (2012)
5. Bishop, C.M.: Pattern recognition and machine learning. springer (2006)
6. Cappé, O., Moulines, E.: On-line expectation–maximization algorithm for latent data models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **71**(3), 593–613 (2009)
7. Chaganty, A.T., Liang, P.: Spectral experts for estimating mixtures of linear regressions. In: ICML. pp. 1040–1048 (2013)
8. Davis, C., Kahan, W.M.: The rotation of eigenvectors by a perturbation. iii. SIAM Journal on Numerical Analysis **7**(1), 1–46 (1970)
9. Ge, R., Huang, F., Jin, C., Yuan, Y.: Escaping from saddle points - online stochastic gradient for tensor decomposition. In: COLT. pp. 797–842 (2015)
10. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American statistical association **58**(301), 13–30 (1963)
11. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent Dirichlet allocation. In: NIPS. pp. 856–864 (2010)
12. Huang, F., Niranjan, U., Hakeem, M.U., Anandkumar, A.: Online tensor methods for learning latent variable models. JMLR **16**, 2797–2835 (2015)
13. Liang, P., Klein, D.: Online EM for unsupervised models. In: NAACL HLT. pp. 611–619. Association for Computational Linguistics (2009)
14. Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in graphical models, pp. 355–368. Springer (1998)
15. Nowozin, S., Lampert, C.H.: Structured learning and prediction in computer vision. Foundations and Trends® in Computer Graphics and Vision **6**(3–4), 185–365 (2011)
16. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
17. Shaban, A., Farajtabar, M., Xie, B., Song, L., Boots, B.: Learning latent variable models by improving spectral solutions with exterior point method. In: UAI. pp. 792–801. AUAI Press (2015)
18. Tung, H.Y., Smola, A.J.: Spectral methods for Indian buffet process inference. In: NIPS. pp. 1484–1492 (2014)
19. Tung, H.Y.F., Wu, C.Y., Zaheer, M., Smola, A.J.: Spectral methods for nonparametric models. arXiv preprint arXiv:1704.00003 (2017)
20. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: ICML. pp. 977–984. ACM (2006)
21. Weyl, H.: Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). Mathematische Annalen **71**(4), 441–479 (1912)
22. Zou, J.Y., Hsu, D.J., Parkes, D.C., Adams, R.P.: Contrastive learning using spectral methods. In: NIPS. pp. 2238–2246 (2013)