SpectralLeader: Online Spectral Learning for Single Topic Models

Abstract

This paper studies how to efficiently learn an optimal latent variable model online from large streaming data. Latent variable models can explain the observed data in terms of unobserved concepts. They are traditionally studied in the unsupervised learning setting, and learned by iterative methods such as the EM. Very few online learning algorithms for latent variable models have been developed, and the most popular one is online EM. Though online EM is computationally efficient, it typically converges to a local optimum. In this work, we motivate and develop SpectralLeader, a novel algorithm to learn latent variable models online from streaming data. SpectralLeader is computationally efficient, and we prove that it quickly learns the global optimum under a bag-of-words model by deriving an $O(\sqrt{n})$ regret bound. Experiment results also demonstrate a consistent performance improvement of SpectralLeader over online EM: in both synthetic and real-world experiments, SpectralLeader's performance is similar with or even better than online EM with optimally tuned parameters.

1 Introduction

Latent variable models are classical approaches to explain observed data via unobserved concepts in supervised and unsupervised learning. They have been successfully applied in a wide variety of fields, such as speech recognition, natural language processing, and computer vision [Rabiner, 1989; Wallach, 2006; Nowozin and Lampert, 2011; Bishop, 2006]. Despite their extensive success, classical latent variable models are limited to the supervised/unsupervised learning setting since they require an available dataset. On the other hand, however, in many practical problems a learning agent needs to learn a latent variable model online while interacting with real-time streaming data with unobserved concepts. For instance, a recommender system would like to learn to cluster its users online based on streaming data recording user impressions and clicks. The goal of this paper is to develop efficient learning algorithms for such online learning problems.

Previous works have proposed several algorithms to learn latent variable models online by extending the classical expectation maximization (EM) algorithm. Those algorithms are known as online EM algorithms, and include the stepwise EM [Cappé and Moulines, 2009; Liang and Klein, 2009] and the incremental EM [Neal and Hinton, 1998]. Similarly as the EM algorithm, each iteration of an online EM algorithm also includes an E-step to fill the values of latent variables based on an estimated distribution, and an M-step to update the model parameters. The main difference is that each step of online EM algorithms only uses currently available data, rather than the whole dataset. This ensures that the online EM algorithms are computationally efficient and can be used to learn a latent variable model online. However, like the EM algorithm, online EM algorithms also have one major drawback: they may converge to local optima and hence suffer from a non-diminishing performance loss.

In this paper, we aim to overcome such limitations by developing a new online learning algorithm that learns the globally optimal latent variable model. Specifically, we propose a novel online learning algorithm, referred to as SpectralLeader, by combining ideas from the state-of-the-art spectral method in unsupervised learning and the follow-the-leader method in online learning. Spectral method [Anandkumar *et al.*, 2014] is recently proposed to learn the parameters for latent variable models in the unsupervised learning setting, with theoretical guarantee of convergence to a global optimum. Specifically, it learns the model parameters by constructing and decomposing high-order tensors of moments. On the other hand, the follow-the-leader method is a classical online learning algorithm [Shalev-Shwartz, 2012]. As is standard in online learning, in this paper we measure the algorithm performance based on the notion of regret, which is the difference of the cumulative loss between our online model at each step and the best model learned in hindsight. We prove that our proposed SpectralLeader algorithm achieves $O(\sqrt{n})$ regret, where n is the number of time steps. This shows that SpectralLeader learns the globally optimal latent variable model efficiently.

The contributions of this paper are fourfold. First, to the best of our knowledge, this is the first paper to formulate the online learning of latent variable models as a regret minimization problem. Moreover, we show that by properly choosing the per-step loss function, the regret in this online learning setting is closely related to the loss function of the spectral method in the unsupervised learning setting. Second, we propose SpectralLeader algorithm, an online learning variant of the spectral method, for the considered online learning problem. SpectralLeader might also exploit reservoir sampling [Vitter, 1985] to reduce its computational and memory complexities. Third, as discussed above, we prove an $O(\sqrt{n})$ regret bound for SpectralLeader. Finally, we compare SpectralLeader and the stepwise EM in extensive synthetic and real-world experiments. Experiment results show that the stepwise EM is very sensitive to its hyperparameter setting; and in all experiments, `SpectralLeader` performs similarly as or even better than the stepwise EM with optimally tuned hyper-parameters.

2 Related Work

This section reviews related work on (i) spectral method for latent variable models and (ii) online learning methods for latent variable models.

The spectral method by tensor decomposition has been widely applied in different latent variable models such as mixtures of tree graphical models [Anandkumar *et al.*, 2014], mixtures of linear regressions [Chaganty and Liang, 2013], hidden Markov models (HMM) [Anandkumar *et al.*, 2012b], latent Dirichlet allocation (LDA) [Anandkumar *et al.*, 2012a], Indian buffet process [Tung and Smola, 2014], and hierarchical Dirichlet process [Tung *et al.*, 2017]. These methods first empirically estimate low-order moments of observations, and then apply decomposition methods (*e.g.*, SVD) to recover the model parameters. The solution of the decomposition is usually unique, such that we can compute a globally optimal solution to those latent variable models.

Traditional online learning methods for latent variable models usually extend the traditional iterative methods for learning latent variable model in the batch setting. Batch EM calculates the sufficient statistics based on all the data [Liang and Klein, 2009]. In online EM, the sufficient statistics can be updated with only recent data in each iteration [Cappé and Moulines, 2009; Neal and Hinton, 1998; Liang and Klein, 2009]. Online variational inference is used to learn LDA efficiently [Hoffman *et al.*, 2010]. Online spectral learning method has also been developed [Huang *et al.*, 2015], with a focus on improving computational efficiency, by conducting optimization of multilinear operations in SGD and avoiding directly forming the tensors. Online stochastic gradient for tensor decomposition has been analyzed in [Ge *et al.*, 2015] with a different online setting. They do not look at the online problem as regret minimization and the analysis focuses on the convergence to the local minimum.

In contrast, in our paper, we develop an online spectral method with a theoretical guarantee of convergence to global optimum. Besides, we discuss designing robust online spectral method in non-stochastic setting where the topics of documents over time are correlated, a setting that has not been studied in the context of online spectral learning [Huang *et al.*, 2015].

3 Spectral Method for Topic Model

In this section, we introduce the background of spectral method in latent variable models. Spectral method works in a wide range of latent variable models; we describe how spectral method works in the simple bag-of-words model [Anandkumar *et al.*, 2014].

In the bag-of-words model, the goal is to understand the latent topic of the documents, based on the observed words in each document. Assume we have K distinct topics, L observed words in each document and the size of the vocabulary is d. This model can be viewed as a mixture model: for the tth document, there is a latent variable Y_t representing the topic and the observed $X_t^{(1)}$, $X_t^{(2)}$, $\cdots$, $X_t^{(L)}$ are conditionally i.i.d. given topic Y_t. Therefore, the parameters of the model include probability of each topic j

$$w_j = P[Y_t = j], \quad j \in [K]. \tag{1}$$

and the conditional probability of each word $u_j \in \mathbb{R}^d$ given topic j, where

$$[u_j]_i = P[X_t^{(l)} = i|Y = j], \quad i \in [d]. \tag{2}$$

When number of observed words $L = 3$, it is enough to construct the third order tensor M_3 as

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

by representing the L words $X_t^{(1)}$, $X_t^{(2)}$, $\cdots$, $X_t^{(L)}$ in the document as d dimensional vector $\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \cdots, \mathbf{x}_t^{(L)} \in \mathbb{R}^d$, where $\mathbf{x}_t^{(l)} = e_i$ if and only if the l-th word in the document is i, $l \in [L]$. Here $e_1, e_2, \cdots e_d$ is the standard coordinate basis for $\mathbb{R}^d$. When $L > 3$, we can construct the M_3, for example, by averaging over all $\binom{L}{3}3!$ ordered triples of words in a document with L words [Anandkumar *et al.*, 2014]. For simplicity, in rest of this paper we discuss the case when $L = 3$ but our algorithm and analysis generalize when $L > 3$. To recover the parameters w_j and u_j, we decompose the third order tensor

$$M_3 = \sum_{i=1}^{K} \omega_i u_i \otimes u_i \otimes u_i. \tag{4}$$

However, in general, obtaining such decomposition for general symmetric tensors (*e.g*, M_3) is NP-hard. Thus, we do not want to deal with general tensors. Instead, we can efficiently obtain the decomposition for orthogonal decomposable tensor, where the eigenvectors are orthogonal. One way to make the tensor M_3 orthogonal decomposable is whitening. We can define whitening matrix as $W = UA^{-1/2}$, where $U \in \mathbb{R}^{d \times K}$ is the matrix of orthonormal eigenvectors of

$$M_2 = \mathbb{E}[\mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)}] = \sum_{i=1}^{K} \omega_i u_i \otimes u_i, \tag{5}$$

and $A \in \mathbb{R}^{K \times K}$ is the diagonal matrix of positive eigenvalues of M_2. To summarize, the spectral method has the following steps [Anandkumar *et al.*, 2014] in Algorithm 1.

4 Online Learning for Topic Models

We study the following online learning problem in a single topic model. Fix a sequence of n document latent topics $(Y_t)_{t=1}^n$, one for each document per time step. The words of the document at time t, X_t, are generated i.i.d. conditioned on Y_t. The sampling distribution of the words is identical at all time steps $t \in [n]$. The goal of the learning agent is to predict a sequence of parameters with low cumulative regret, with respect to the best solution in hindsight of knowing $(Y_t)_{t=1}^n$ and $(X_t)_{t=1}^n$.

Algorithm 1: Spectral method for latent bag-of-words model.

```
Data: One hot encoding $(\mathbf{x}_t^{(1)})_{t=1}^n$, $(\mathbf{x}_t^{(2)})_{t=1}^n$ and
      $(\mathbf{x}_t^{(3)})_{t=1}^n$ for data $(X_t^{(1)})_{t=1}^n$, $(X_t^{(2)})_{t=1}^n$ and
      $(X_t^{(3)})_{t=1}^n$.
Result: The latent variable model $\omega_i$ and $u_i$, where
        $i \in [K]$.
1 $M_2 = \frac{1}{n}\sum_{t=1}^n \mathbf{x}_t^{(1)} \otimes \mathbf{x}_t^{(2)}$
2 $W = UA^{-1/2}$ {$U \in \mathbb{R}^{d\times K}$ is the matrix of orthonormal
    eigenvectors of $M_2$. $A \in \mathbb{R}^{K\times K}$ is the diagonal matrix
    of positive eigenvalues of $M_2$ }.
3 $\mathbf{y}_t^{(1)} = W^\top \mathbf{x}_t^{(1)}, \mathbf{y}_t^{(2)} = W^\top \mathbf{x}_t^{(2)}, \mathbf{y}_t^{(3)} = W^\top \mathbf{x}_t^{(3)}$
4 $T = \frac{1}{n}\sum_{t=1}^n \mathbf{y}_t^{(1)} \otimes \mathbf{y}_t^{(2)} \otimes \mathbf{y}_t^{(3)}$
5 Decompose $T$ by power iteration to get $\lambda_i$ and $v_i$:
    $T = \sum_{i=1}^K \lambda_i v_i \otimes v_i \otimes v_i$
6 $\omega_i = \frac{1}{\lambda_i^2}$
7 $u_i = \lambda_i (W^\top)^+ v_i$
```

To simplify notation and reduce clutter, we introduce the following notation. We denote by T_t the tensor constructed from the first t observations, in line 4 of Algorithm 1; and by $\bar{T}_t$ its expectation with respect to random observations $(X_s)_{s=1}^t$. We refer to tensor decompositions in line 5 of Algorithm 1 by $\theta = ((\lambda_i)_{i=1}^K, (v_i)_{i=1}^K)$ and to the corresponding estimated tensor by $T(\theta) = \sum_{i=1}^K \lambda_i v_i \otimes v_i \otimes v_i$. When the tensor decomposition is computed from the first t observations, we denote it by $\theta_t = f(T_t)$, where f is a mapping from tensor T_t to θ_t in line 5 of Algorithm 1.

Given a model θ, we define the *cumulative loss* over the first t steps as

$$\mathcal{L}_t(\theta) = t\|\bar{T}_t - T(\theta)\|^p\,, \tag{6}$$

where $\|\cdot\|$ is the *tensor spectral norm* in Anandkumar *et al.* [2014] and $p \in \mathbb{Z}_+$ can be any positive integer. However, to develop online algorithm with sublinear regret, we should choose $p > 1$. We set $p = 2$ in (6) and will have detailed discussions in Theorem 1 and 2. This loss is the same as the offline batch loss of the spectral method on the data in the first t steps [Anandkumar *et al.*, 2014]. We define the *loss* at time t as

$$\begin{aligned}\ell_t(\theta) &= \mathcal{L}_t(\theta) - \mathcal{L}_{t-1}(\theta) &(7)\\ &= t\|\bar{T}_t - T(\theta)\|^2 - (t-1)\|\bar{T}_{t-1} - T(\theta)\|^2\end{aligned}$$

and assume that $\|\bar{T}_0 - T(\theta)\| = 0$. The reason for the above formulation is that the sum of the first n losses, which is

$$\sum_{t=1}^n \ell_t(\theta) = n\|\bar{T}_n - T(\theta)\|^2\,, \tag{8}$$

is minimized by Algorithm 1. This definition of the loss function $\ell_t(\theta)$ is motivated by the work of Kar *et al.* [2014] on non-decomposable loss functions. Roughly speaking, $\ell_t(\theta)$ measures the extra loss due to including the observation at time t, X_t, when T_{t-1} is updated to T_t.

Our goal is to bound the cumulative regret

$$R(n) = \sum_{t=1}^n \ell_t(\hat{\theta}_{t-1}) - \sum_{t=1}^n \ell_t(\theta_n)\,, \tag{9}$$

where $\hat{\theta}_{t-1}$ is the *solution of the learning agent at time* t and θ_n is the *best solution in hindsight*.

5 Algorithm SpectralLeader

We propose SpectralLeader, an online learning algorithm for minimizing the regret in a single topic model, which is defined in (9). At time t, it chooses solution

$$\hat{\theta}_{t-1} = f(T_{t-1})\,, \tag{10}$$

where T_{t-1} is the tensor constructed from the first $t-1$ observations, in line 4 of Algorithm 1.

SpectralLeader is not computationally efficient because its time complexity at time t depends on time t. In particular, the whitening operation in line 3 of Algorithm 1 depends on t because all past observations are whitened by a whitening matrix W that changes with t.

A more memory and computationally efficient algorithm can be designed using *reservoir sampling*. Originally, $\bar{T}_{t-1}$ is calculated based on all observations from the first $t-1$ steps. Instead, we maintain a pool of R observations. When $t < R$, a new observation is added to the pool. When $t \geq R$, a new observation replaces a random observation in the pool with probability R/t. Then we can approximate T_{t-1} by calculating it only based on the observations in the pool. With reservoir sampling, SpectralLeader has per-step memory and computational cost independent of t.

6 Analysis

In this section, we derive bounds on the n-step regret of SpectralLeader. We analyze two cases. In the first case, we assume the learning agent observes expected tensors, $\bar{T}_{t-1}$ at each time t. In this noise-free case, we derive a $O(\log n)$ bound on the n-step regret of SpectralLeader. In the second case, the learning agent is assumed to observe empirical tensors T_{t-1}, which are noisy realization of $\bar{T}_{t-1}$. The noises come from the differences between the empirically observed tensors and the expected tensors. In this noisy case, we derive a $O(\sqrt{n})$ bound on the n-step regret of SpectralLeader.

At time t, the observed tensor can be represented as $T_t = \frac{1}{t}\sum_{z=1}^t Y_{t,z}$, where $Y_{t,z} = W_t^\top \mathbf{x}_z^{(1)} \otimes W_t^\top \mathbf{x}_z^{(2)} \otimes W_t^\top \mathbf{x}_z^{(3)}$, and W_t is the whitening matrix. Similarly, the expected tensor can be represented as $\bar{T}_t = \frac{1}{t}\sum_{z=1}^t \bar{Y}_{t,z}$, where $\bar{Y}_{t,z} = \mathbb{E}[Y_{t,z}]$. Tensor $\bar{T}_t$ is symmetric and has orthogonal decomposition $\bar{T}_t = \sum_{i=1}^K \lambda_{t,i} v_{t,i} \otimes v_{t,i} \otimes v_{t,i}$. We define $\lambda_{t,\min} = \min\{\lambda_{t,i} : i \in [K]\}$, and define $\lambda_{\min} = \min\{\lambda_{t,\min} : t \in [n]\}$. We further define the event

$$\mathcal{E} = \{\forall t \in [n] : \|\bar{T}_t - T_t\| \leq c_1\sqrt{\frac{1}{t}}\},$$

where c_1 is a constant.

Before the regret analysis, we make the following assumptions. We show that they are reasonable in Section 6.4.

Assumption 1. *We assume that at each time t, each entry in $Y_{t,z}$ is bounded. That is,*

$$\forall t, z, i, j, k : |Y_{t,z}(i,j,k)| \le r_1,$$

where $r_1 \in \mathbb{R}$ is a constant and $z \in [t]$.

This assumption indicates that after whitening, the input one-hot encoding does not change wildly.

Assumption 2. *We assume that at each time t,*

$$\forall t, z, i, j, k : |Y_{t,z}(i,j,k) - Y_{t-1,z}(i,j,k)| \le \frac{r_2}{t},$$

where $r \in \mathbb{R}$ is a constant and $z \in [t-1]$.

This assumption indicates that the whitening matrices are stable, in the sense that they do not change too much between two consequent steps.

6.1 Noise-Free Case

We first analyze the regret in the case when expected tensor $\bar{T}_{t-1}$ are observed at each time t.

Theorem 1. *When the learning agent observes expected tensor $\bar{T}_{t-1}$ at each time t, for any sequence of $(\bar{T}_t)_{t=1}^n$,*

$$\mathbb{E}\left[R(n)\right] \le c_2^2 \log n,$$

for some $c_2 \ge 0$ independent of n.

Proof. Since $\bar{T}_{t-1}$ is decomposable, the agent learns the solution $f(\bar{T}_{t-1})$ and the loss at time t is

$$\begin{aligned}
&\ell(f(\bar{T}_{t-1})) \\
&= t\|\bar{T}_t - T(f(\bar{T}_{t-1}))\|^2 - (t-1)\|\bar{T}_{t-1} - T(f(\bar{T}_{t-1}))\|^2 \\
&= t\|\bar{T}_t - \bar{T}_{t-1}\|^2 - (t-1)\|\bar{T}_{t-1} - \bar{T}_{t-1}\|^2 \\
&\le t(c_2 t^{-1})^2 \\
&= c_2^2 t^{-1},
\end{aligned} \tag{11}$$

where the inequality uses the Lemma 1, in which we show that the difference of expected tensors between two consequent steps is bounded. Therefore, the cumulative regret in n steps is bounded as $\mathbb{E}\left[R(n)\right] \le \sum_{t=1}^n \ell(f(\bar{T}_{t-1})) \le c_2^2 \log n$. ■

In this ideal case where we can observe expected tensors, `SpectralLeader` has $O(\log n)$ regret, which means that `SpectralLeader` can quickly find the solutions as good as the best solution in hindsight. We further derive the regret bound in a realistic case in Section 6.2, where we observe empirical tensors. Remember the cumulative loss is defined as $\|\bar{T}_t - T(\theta)\|^p$ in (6). With this definition, the loss at time t in (11) is $c_2^2 t^{1-p}$. To develop online algorithm with sublinear regret, we should choose $p > 1$ and thus we set $p = 2$ in (6).

6.2 Noisy Case

We further analyze the regret in the case when empirical tensor T_{t-1} are observed at each time t.

Theorem 2. *When the learning agent observes empirical tensor T_{t-1} at each time t, for any sequence of $(\bar{T}_t)_{t=1}^n$,*

$$\begin{aligned}
\mathbb{E}\left[R(n)\right] \le 2\, c_4 K^3 + c_5 \left(\frac{c_1^2 K^2}{c_3^2 \lambda_{min}^2} - 1\right) \\
+ 110\, c_1 c_2 \sqrt{n} + c_2^2 \log n + 3025\, c_1^2 \log n
\end{aligned}$$

in expectation over $(T_t)_{t=1}^n$, for some $c_1, c_2, c_3, c_4, c_5 \ge 0$ independent of n.

Proof. At each time t, learning agent observes T_t, which is a noisy realization of $\bar{T}_t$. To analyze the upper bound of regret, we discuss three possible cases.

First, $\mathcal{E}$ does not hold, in at least one time t. Lemma 2 bounds the difference between the expected tensor and empirical tensor at each time t. According to Lemma 2, the regret introduced in the first case is bounded by $2\frac{K^3}{n} c_4 n = 2 c_4 K^3$, where $2\frac{K^3}{n}$ is the probability of $\mathcal{E}$ does not hold, and the constant c_4 is the upper bound of the regret in each time when $\mathcal{E}$ does not hold.

Second, $\mathcal{E}$ holds and $t < t_0$. Lemma 3 bounds the difference between the expected tensor and the tensor reconstructed from the estimated parameters by `SpectralLeader`. According to Lemma 3, the regret introduced in the second case is bounded trivially by $c_5(t_0 - 1) = c_5(\frac{c_1^2 K^2}{c_3^2 \lambda_{\text{min}}^2} - 1)$, where constant c_5 is the upper bound of the regret in each time before t_0.

Third, $\mathcal{E}$ holds and $t \ge t_0$. To reduce clutter, we define $\alpha = \|\bar{T}_t - T(f(T_{t-1}))\|$ and $\beta = \|\bar{T}_{t-1} - T(f(T_{t-1}))\|$. Thus, the regret introduced by the third case is $\sum_{t=1}^n \ell_t(f(T_{t-1}))$. We bound the regret as

$$\begin{aligned}
\ell_t(f(T_{t-1})) &= t(\alpha^2 - \beta^2) + \beta^2 \\
&= t(\alpha - \beta)(\alpha + \beta) + \beta^2 \\
&\le t\|\bar{T}_t - \bar{T}_{t-1}\|(2\beta + \|\bar{T}_t - \bar{T}_{t-1}\|) + \beta^2 \\
&\le t c_2 t^{-1}(110 c_1 (t-1)^{-1/2} + c_2 t^{-1}) + \\
&\quad (55\, c_1 (t-1)^{-1/2})^2 \\
&= 110\, c_1 c_2 (t-1)^{-1/2} + c_2^2 t^{-1} + 3025\, c_1^2 (t-1)^{-1},
\end{aligned}$$

where the first inequality is based on reverse triangle inequality that $\alpha - \beta \le \|\bar{T}_t - \bar{T}_{t-1}\|$ and $\alpha \le \beta + \|\bar{T}_t - \bar{T}_{t-1}\|$, and the second inequality is based on $\|\bar{T}_t - T(f(T_t))\| \le 55\, c_1 \sqrt{\frac{1}{t}}$ by Lemma 3 and $\|\bar{T}_t - \bar{T}_{t-1}\| \le c_2 t^{-1}$ by Lemma 1.

By adding up all the introduce regret from the three cases, the cumulative regret in n steps

$$\begin{aligned}
\mathbb{E}\left[R(n)\right] \le \quad & \underbrace{2 c_4 K^3}_{\text{regret by the first case}} + \underbrace{c_5 \left(\frac{c_1^2 K^2}{c_3^2 \lambda_{\text{min}}^2} - 1\right)}_{\text{regret by the second case}} \\
& + \underbrace{110\, c_1 c_2 \sqrt{n-1} + c_2^2 \log n + 3025\, c_1^2 \log(n-1)}_{\text{regret by the third case}} \\
\le \quad & 2\, c_4 K^3 + c_5 \left(\frac{c_1^2 K^2}{c_3^2 \lambda_{\text{min}}^2} - 1\right) \\
& + 110\, c_1 c_2 \sqrt{n} + c_2^2 \log n + 3025\, c_1^2 \log n,
\end{aligned}$$

and our claim follows. ■

Similarly, if we set $p = 1$ in the cumulative loss in (6), $\ell_t(f(T_{t-1}))$ is on the order of a constant and `SpectralLeader` can not achieve sublinear regret.

6.3 Technical Lemmas

Lemma 1. *For any t, the expected tensors at time t and $t-1$ are close to each other. Formally,*

$$\exists c_2, \forall t : \|\bar{T}_t - \bar{T}_{t-1}\| \leq c_2 t^{-1}$$

Proof. Assume $|Y_{t,z}(i,j,k)| \leq r_1$ and $|Y_{t,z}(i,j,k) - Y_{t-1,z}(i,j,k)| \leq \frac{r_2}{t}$ hold in Assumption 1 and 2, we have

$$\begin{aligned}
&\|\bar{T}_t - \bar{T}_{t-1}\| \\
&= \|\frac{1}{t}\sum_{z=1}^{t}\bar{Y}_{t,z} - \frac{1}{t-1}\sum_{z=1}^{t-1}\bar{Y}_{t-1,z}\| \\
&= \frac{1}{t(t-1)}\|(t-1)\sum_{z=1}^{t}\bar{Y}_{t,z} - t\sum_{z=1}^{t-1}\bar{Y}_{t-1,z}\| \\
&= \frac{1}{t(t-1)}\|t\sum_{z=1}^{t}\bar{Y}_{t,z} - \sum_{z=1}^{t}\bar{Y}_{t,z} - t\sum_{z=1}^{t-1}\bar{Y}_{t-1,z}\| \\
&= \frac{1}{t(t-1)}\|t\bar{Y}_{t,t} - \sum_{z=1}^{t}\bar{Y}_{t,z} + t\sum_{z=1}^{t-1}(\bar{Y}_{t,z} - \bar{Y}_{t-1,z})\| \\
&= \frac{1}{t(t-1)}\|(t-1)\bar{Y}_{t,t-1} - \sum_{z=1}^{t-1}\bar{Y}_{t,z} + t\sum_{z=1}^{t-1}(\bar{Y}_{t,z} - \bar{Y}_{t-1,z})\| \\
&\leq \frac{1}{t}\left(\|\bar{Y}_{t,t-1}\| + \frac{\|\sum_{z=1}^{t-1}\bar{Y}_{t,z}\|}{t-1} + \frac{\|t\sum_{z=1}^{t-1}(\bar{Y}_{t,z} - \bar{Y}_{t-1,z})\|}{t-1}\right) \\
&\leq \frac{1}{t}\left(\|\bar{Y}_{t,t-1}\|_F + \frac{\|\sum_{z=1}^{t-1}\bar{Y}_{t,z}\|_F}{t-1} + \frac{\|t\sum_{z=1}^{t-1}(\bar{Y}_{t,z} - \bar{Y}_{t-1,z})\|_F}{t-1}\right) \\
&\leq \frac{1}{t}\left(\sqrt{K^3}r_1 + \sqrt{K^3}r_1 + \sqrt{K^3}r_2\right).
\end{aligned}$$

By taking $c_2 = 2\sqrt{K^3}r_1 + \sqrt{K^3}r_2$, our claim holds. ■

Lemma 2. *For any sequence of $(\bar{T}_t)_{t=1}^n$, $\mathcal{E}$ holds with at least probability $1 - 2\frac{K^3}{n}$, where $c_1 = \frac{K^3 r_1}{\sqrt{\log n}}$.*

Proof. Let $\bar{\mathcal{E}}$ be the complement of event $\mathcal{E}$. That is,

$$\bar{\mathcal{E}} = \{\exists t \in [n] : \|\bar{T}_t - T_t\| > c_1\sqrt{\frac{1}{t}}\}.$$

Since $\|\bar{T}_t - T_t\| \leq \|\bar{T}_t - T_t\|_F$, we have

$$\begin{aligned}
&P(\bar{\mathcal{E}}) \\
&\leq P\left(\exists t : \|\bar{T}_t - T_t\|_F > c_1\sqrt{\frac{1}{t}}\right) \\
&\leq P\left(\exists t, i, j, k : |\bar{T}_t(i,j,k) - T_t(i,j,k)| > \frac{c_1}{K^3}\sqrt{\frac{1}{t}}\right).
\end{aligned}$$

Here $|T_t(i,j,k)| \leq r_1$, because $|Y_{t,l}(i,j,k)| \leq r_1$. According to Azuma-Hoeffding inequality,

$$P\left(|\bar{T}_t(i,j,k) - T_t(i,j,k)| > \frac{c_1}{K^3}\sqrt{\frac{1}{t}}\right) \leq 2e^{-2\frac{c_1^2}{K^6 r_1^2}\log n}.$$

Therefore,

$$\begin{aligned}
&P(\mathcal{E}) \\
&= 1 - P(\bar{\mathcal{E}}) \\
&\geq 1 - P\left(\exists t, i, j, k : |\bar{T}_t(i,j,k) - T_t(i,j,k)| > \frac{c_1}{K^3}\sqrt{\frac{1}{t}}\right) \\
&\geq 1 - \sum_{i,j,k,t} P\left(|\bar{T}_t(i,j,k) - T_t(i,j,k)| > \frac{c_1}{K^3}\sqrt{\frac{1}{t}}\right) \\
&\geq 1 - 2nK^3 e^{-2\frac{c_1^2}{K^6 r_1^2}\log n},
\end{aligned}$$

where the second inequality uses union bound. Let $c_1 = \frac{K^3 r_1}{\sqrt{\log n}}$, we have $P(\mathcal{E}) \geq 1 - 2\frac{K^3}{n}$ and our claim follows. ■

In case when $\mathcal{E}$ holds, we show the following Lemma.

Lemma 3. *Given that $\mathcal{E}$ holds holds, for any sequence of $(\bar{T}_t)_{t=1}^n$,*

$$\|\bar{T}_t - T(f(T_t))\| \leq 55\, c_1\sqrt{\frac{1}{t}}$$

holds after time $t_0 = \frac{c_1^2 K^2}{c_3^2 \lambda_{min}^2}$, where c_3 is a universal constant.

Proof. Given that $\mathcal{E}$ holds, $\|\bar{T}_t - T_t\| \leq c_1\sqrt{\frac{1}{t}}$ at each time t. Let

$$\epsilon_t = c_1\sqrt{\frac{1}{t}} \leq c_3\frac{\lambda_{\text{min}}}{K}. \tag{12}$$

We have

$$\|\bar{T}_t - T_t\| \leq \epsilon_t, \text{ and } \epsilon_t \leq c_3\frac{\lambda_{t,\text{min}}}{K}.$$

From Theorem 5.1 of Anandkumar *et al.* [2014], we have

$$\|\bar{T}_t - T(f(T_t))\| \leq 55\,\epsilon_t.$$

Therefore,

$$\|\bar{T}_t - T(f(T_t))\| \leq 55\, c_1\sqrt{\frac{1}{t}}.$$

When $t \geq \frac{c_1^2 K^2}{c_3^2 \lambda_{\text{min}}^2}$, we can always find a ϵ_t, such that (12) holds. Therefore, after $\frac{c_1^2 K^2}{c_3^2 \lambda_{\text{min}}^2}$ steps, we always have $\|\bar{T}_t - T(f(T_t))\| \leq 55\sqrt{\frac{1}{t}}$. ■

6.4 Discussion

In this section, we show Assumption 1 and 2 are reasonable.

We show each entry in $Y_{t,z}$ is bounded in Assumption 1. At time t, the observed second order tensor is $M_{2,t}$ and the expected second order tensor is $\bar{M}_{2,t}$. From Anandkumar *et al.* [2014], we can take whitening matrix as $W_t = U_t A_t^{-\frac{1}{2}}$, where $U_t \in \mathbb{R}^{d\times K}$ is the matrix of orthonormal eigenvectors of $M_{2,t}$, and $A_t \in \mathbb{R}^{k\times k}$ is the diagonal matrix of positive eigenvalues of $M_{2,t}$. With whitening, $\mathbf{y}_z^{(l)} = U_t A_t^{-\frac{1}{2}} \mathbf{x}_z^{(l)}$.

To bound each entry in $Y_{t,z}$, we bound each entry in $\mathbf{y}_z^{(l)}$. To achieve this, we bound each entry in $\mathbf{x}_z^{(l)}$, U_t, and $A_t^{-\frac{1}{2}}$, respectively. According to the definition, each entry in $\mathbf{x}_z^{(l)}$ and U_t is bounded by 1. Denote the smallest eigenvalue of $\bar{M}_{2,t}$ as $\lambda_d(\bar{M}_{2,t})$ and the smallest eigenvalue of $M_{2,t}$ as $\lambda_d(M_{2,t})$. Then, each diagonal entry in $A_t^{-\frac{1}{2}}$ is upper bounded by $\frac{1}{\sqrt{\lambda_d(M_{2,t})}}$. Therefore, each entry in $\mathbf{y}_z^{(l)} = U_t A_t^{-\frac{1}{2}} \mathbf{x}_z^{(l)}$ is upper bounded by $\frac{1}{\sqrt{\lambda_d(M_{2,t})}}$. As a result, $|Y_{t,z}(i,j,k)| \le \frac{1}{\lambda_d(M_{2,t})^{3/2}}$.

To have finite upper bound for $|Y_{t,z}(i,j,k)|$, we should have $\lambda_d(M_{2,t}) > 0$. By Weyl's inequality, we have

$$|\lambda_d(\bar{M}_{2,t}) - \lambda_d(M_{2,t})| \le \|\bar{M}_{2,t} - M_{2,t}\|.$$

Therefore,

$$\lambda_d(M_{2,t}) \ge -\|\bar{M}_{2,t} - M_{2,t}\| + \lambda_d(\bar{M}_{2,t}) > 0.$$

To achieve this, one sufficient condition is

$$\begin{aligned} &\|\bar{M}_{2,t} - M_{2,t}\|_F \\ &= \sqrt{\sum_{i,j,k} (\bar{M}_{2,t}(i,j,k) - M_{2,t}(i,j,k))^2} \\ &= \sqrt{d^3 (\bar{M}_{2,t}(i,j,k) - M_{2,t}(i,j,k))^2} < \lambda_d(\bar{M}_{2,t}). \end{aligned}$$

That is,

$$|\bar{M}_{2,t}(i,j,k) - M_{2,t}(i,j,k)| < d^{-3/2}\lambda_d(\bar{M}_{2,t}).$$

According to the Azuma-Hoeffding inequality, We have

$$\begin{aligned} &P\left(|\bar{M}_{2,t}(i,j,k) - M_{2,t}(i,j,k)| > d^{-3/2}\lambda_d(\bar{M}_{2,t})\right) \\ &> 1 - 2e^{-2td^{-3}\lambda_d^2(\bar{M}_{2,t})}. \end{aligned}$$

That is, with probability $1 - 2e^{-2td^{-3}\lambda_d^2(\bar{M}_{2,t})}$, $\lambda_d(M_{2,t}) > 0$ and $|Y_t(i,j,k)|$ has a finite upper bound $\frac{1}{\lambda_d(M_{2,t})^{3/2}}$.

7 Experiments

In this section, we evaluate `SpectralLeader` and compare it with state-of-the-art baselines in multiple numerical experiments. Specifically, we demonstrate experimental results in synthetic problems with both stochastic and non-stochastic settings, as well as two problems based on large-scale real-world datasets.

X	$P(X\|Y=1)$	$P(X\|Y=2)$	$P(X\|Y=3)$
$X=1$	p	$\frac{1-p}{2}$	$\frac{1-p}{2}$
$X=2$	$\frac{1-p}{2}$	p	$\frac{1-p}{2}$
$X=3$	$\frac{1-p}{2}$	$\frac{1-p}{2}$	p

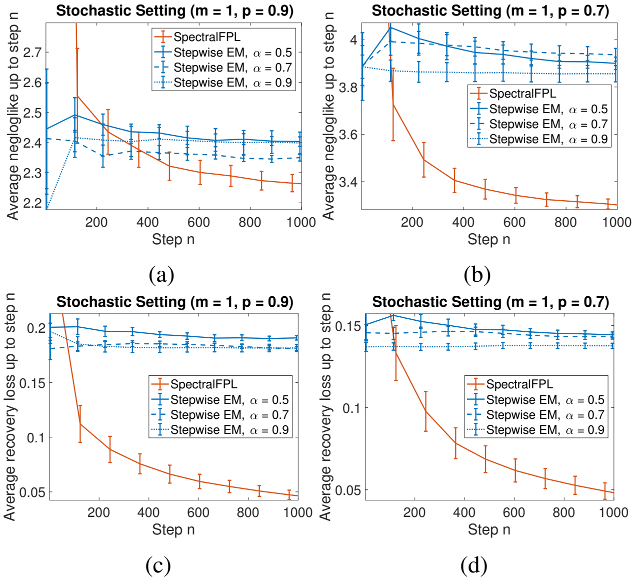
Table 1: The conditional distribution of words in the synthetic problems.

Our chosen baseline is stepwise EM [Cappé and Moulines, 2009], an online EM algorithm. We choose this baseline as previous work shows that it outperforms other online EM algorithms, such as incremental EM [Liang and Klein, 2009]. As detailed in [Liang and Klein, 2009; Cappé and Moulines, 2009], stepwise EM has two key tuning parameters: the step-size reduction power α and the mini-batch size m. In particular, the learning rates in stepwise EM decrease with α. In the following experiments, we compared `SpectralLeader` to stepwise EM with various α and m.

To have a fair comparison, we define two metrics: (i) *negative predictive log likelihood up to step* n, $\mathcal{L}_n^{(1)} = \sum_{t=2}^n \left(-\log \sum_{i=1}^K P_{\theta_{t-1}}(Y=i) \prod_{l=1}^L P_{\theta_{t-1}}(X = X_t^{(l)}|Y=i)\right)$ and (ii) *recovery error up to step* n, $\mathcal{L}_n^{(2)} = \sum_{t=2}^n \|M_2(\theta^*) - M_2(\theta_{t-1})\|_F^2$, where $M_2(\theta^*)$ and $M_2(\theta_t)$ are the reconstructed second order moments (reconstructed by (5)) from optimal model θ^* and learned model θ_t, respectively. Here $\|\cdot\|_F$ is the Frobenius norm. We choose $\mathcal{L}_n^{(2)}$ as metric instead of $\mathcal{L}_n$ in (6). Because metric $\mathcal{L}_n^{(2)}$ can be easily computed for both `SpectralLeader` and stepwise EM. This metric measures parameter reconstruction error, and therefore is also closely related to the objective of the spectral method. In synthetic problems, we know θ^*. In real-world problems, we learn θ^* by the spectral method because we have all data in advance. Note that the EM in batch setting minimizes the negative log likelihood, but `SpectralLeader` in batch setting minimizes the recovery error of tensors. In the following experiments, at step n, we report (i) *average negative log likelihood up to step* n, which is $\frac{1}{n}\mathcal{L}_n^{(1)}$ and (ii) *average recovery error up to step* n, which is $\frac{1}{n}\mathcal{L}_n^{(2)}$. All the reported results are averaged over 10 runs.

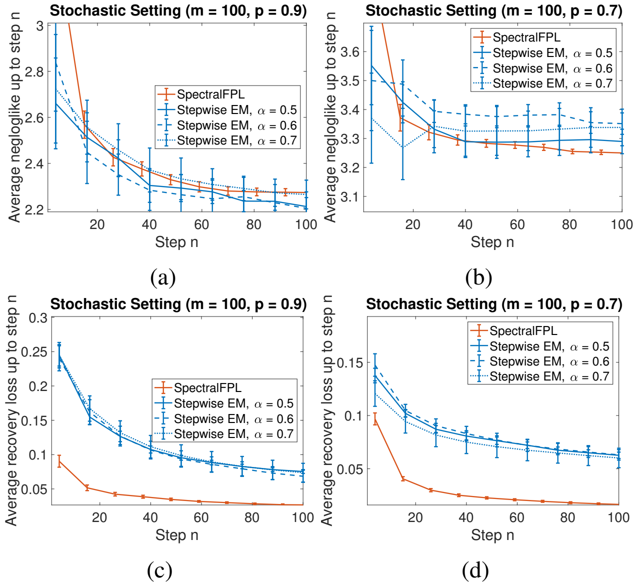
7.1 Synthetic Stochastic Setting

We compare our methods with stepwise EM on two synthetic problems in the stochastic setting. In this setting, the topic of the documents at each t is sampled from a fixed distribution. This setting is to simulate the scenario where there is no correlation between the topics in a sequence of documents. We set number of distinct topics $K = 3$, number of observed words $L = 3$ in each document, and the size of the vocabulary $d = 3$. Considering in practice, some topics are more popular than other topics, we set the following distribution for topic Y. At each step, the topic is randomly sampled from the distribution where $P(Y = 1) = 0.15$, $P(Y = 2) = 0.35$, and $P(Y = 3) = 0.5$. Given each topic, the conditional probability of words is listed in Table 1. We evaluate on easy problems where $p = 0.9$, and then evaluate on more difficult problems



(a) (b) (c) (d)

Figure 1: The comparisons between SpectralLeader and stepwise EM on synthetic problems under stochastic setting, when $m = 1$. The x-axis is step n. In the first row the y-axis is the average negative log likelihood up to step n, while in the second row the y-axis is the average recovery error up to step n.



(a) (b) (c) (d)

Figure 2: The comparisons between SpectralLeader and stepwise EM on synthetic problems under stochastic setting, when $m = 100$. The x-axis is step n. In the first row the y-axis is the average negative log likelihood up to step n, while in the second row the y-axis is the average recovery error up to step n.

where $p = 0.7$. With smaller p, the conditional distribution of words given different topic becomes similar, the difficulty of distinguishing different topics increases. For stepwise EM, the main tunable parameter is α. The smaller the α, the more quickly the old sufficient statistics are forgotten.

We show that stepwise EM is very sensitive to its parameter setting α, while SpectralLeader is competitive or even better, compared to the stepwise EM with its best α in Figure 1. It can be observed that, for stepwise EM in different problems, the best α leading to the lowest averaged negative log likelihood or recovery loss are different. For example, the best α is 0.7 in Figure 1a, and the best α is 0.9 in Figure 1b. But, in all cases, SpectralLeader performs the best.

We further show competitive results of SpectralLeader when the mini-batch size $m = 100$, where stepwise EM usually improves [Liang and Klein, 2009]. In this case, 100 documents are used to calculate the sufficient statistics for each update of stepwise EM. We show the results of 100 steps before different methods converge. Our results are shown in Figure 2. Indeed, we observe stepwise EM improves when m is increased from 1 to 100. But, still SpectralLeader performs better or competitive, compared to stepwise EM with its best α, except Figure 2a. Another observation is that stepwise EM is again sensitive. Even for the same problem, with different m, the best α of stepwise EM is different. As an instance, the best α is 0.9 in Figure 1b, while the best α is 0.5 in Figure 2b.

These results indicate that the best parameters of stepwise EM depend on the individual problem, and a careful grid search of α and m is usually needed to optimize stepwise EM. However, in practice a grid search in online learning is almost impossible: we can not see all the data in advance, to select best parameters for stepwise EM in the online setting.

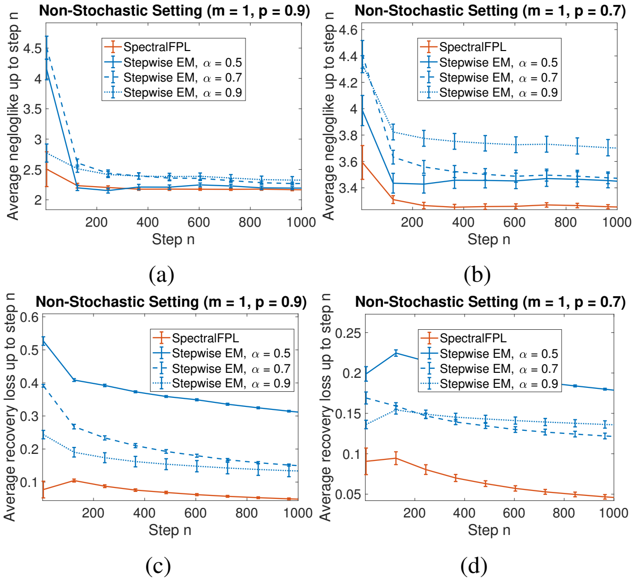
7.2 Synthetic Non-stochastic Setting

We evaluate different methods on two synthetic problems in the non-stochastic setting, where the topic of the documents at each t is correlated. This setting is the same as stochastic setting, except that topics of the documents are strongly correlated in the streaming data. We look at an extreme case of correlated topics in the steaming data. For each 100 steps, sequentially we have 15 documents from topic 1, 35 documents from topic 2, and 50 documents from topic 3. We compare SpectralLeader and stepwise EM in this non-stochastic setting.

In Figure 3, we show the competitive performance of SpectralLeader in non-stochastic setting. First, for stepwise EM, the α leading to lowest negative log likelihood is 0.5. This result well matches the fact that the smaller the α, the more quickly the old sufficient statistics is forgotten, and more stepwise EM adapts to the non-stochastic setting. Second, in terms of adaptation to correlated topics in non-stochastic setting, SpectralLeader is even better than stepwise with $\alpha = 0.5$. Note that 0.5 is the smallest valid value of $\alpha = 0.5$ for stepwise EM [Liang and Klein, 2009].

7.3 Real World Problems

In this section, we compare our SpectralLeader to stepwise EM on real world problems. We evaluate on both NYTimes news articles and PubMed abstracts [Lichman, 2013], which are popularly used in the evaluation of bag-of-words model [Huang *et al.*, 2015; Lichman, 2013]. As a preprocessing step, we retain the top 500 frequent words in the vocabulary. We set $K = 5$. We evaluate our online learning experiments on the scale of 100K documents. In order to do this, we filter out the document with less than 50 words. For those



(a) (b)

(c) (d)

Figure 3: The comparisons between `SpectralLeader` and stepwise EM under non-stochastic setting, when $m = 1$. The x-axis is step n. In the first row the y-axis is the average negative log likelihood up to step n, while in the second row the y-axis is the average recovery error up to step n.

documents with more than 50 words, we randomly uniform down sample 50 words for each document. This downsampling is to ensure that in each step (or mini-batch) of online learning, there is the same amount of words.

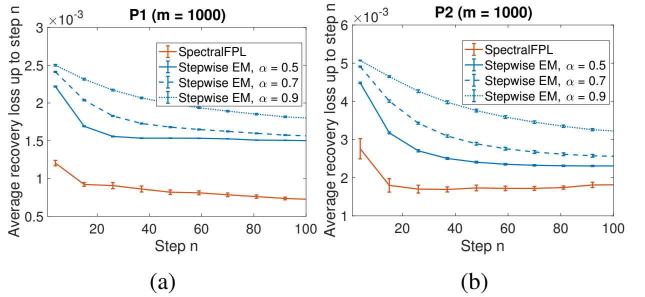
We compare `SpectralLeader` to stepwise EM with different α, when mini-batch size is $m = 1$, 100 and 1000. The setting of $m = 1000$ is to evaluate the performance of `SpectralLeader` with *reservoir sampling* on large scale streaming data with 5M words. We set the window size of reservoir sampling to 10000. We show the competitive results of `SpectralLeader` on real world datasets with various α and m in Table 2 and Figure 4. We show the recovery error curves of different algorithms when $m = 1000$ in Figure 4. When $m = 1$ and 100, actually the curves are similar to the ones when $m = 1000$. We summarize results for various m in Table 2. For $m = 1$, we report both of our metrics at $n = 1000$, and for $m = 100$ and $m = 1000$, we report both of our metrics at $n = 100$.

Figure 4 showed that for recovery error, `SpectralLeader` performs better than stepwise EM with its best α for both real world datasets. When we downsample data with window size 10000 by reservoir sampling, the model still learns, as shown in Figure 4. Note that at each step n, we process $m = 1000$ documents. When $n > 10$, the data starts to be downsampled by reservoir sampling. We observe that the curves of `SpectralLeader` still decrease when $n > 10$.

Similar comparative behavior is observed in Table 2: `SpectralLeader` performs better than stepwise EM with best α and across different m in terms of parameter recovery loss. In terms of negative predictive log-likelihood, `SpectralLeader` is not as good as the stepwise EM with it best parameter setting. However, directly using `SpectralLeader` without the effort of tuning any parameters can still provide reasonably good performance under negative log-likelihood. `SpectralLeader` usually performs much better than the stepwise EM with its worse parameter setting. Thus, even under negative log likelihood metric, `SpectralLeader` is very useful in practice: we can quickly build reasonable baseline results by `SpectralLeader` without any parameter tuning.

		Avg Neg Loglik		Avg Rec Loss	
m	Methods	P1	P2	P1	P2
1	`SpectralLeader`	310.7151	299.0675	**0.0021**	**0.0032**
	Stepwise EM, $\alpha = 0.5$	314.3184	314.3601	0.0031	0.0049
	Stepwise EM, $\alpha = 0.7$	305.2978	297.0082	0.0024	0.0042
	Stepwise EM, $\alpha = 0.9$	**303.9964**	**293.1403**	0.0022	0.0040
100	`SpectralLeader`	302.5069	287.3493	**0.0011**	**0.0018**
	Stepwise EM, $\alpha = 0.5$	**296.7833**	**285.9445**	0.0014	0.0022
	Stepwise EM, $\alpha = 0.7$	297.0398	286.6765	0.0014	0.0024
	Stepwise EM, $\alpha = 0.9$	300.6628	290.7802	0.0016	0.0031
1000	`SpectralLeader`	300.6544	287.0066	**0.0007**	**0.0018**
	Stepwise EM, $\alpha = 0.5$	**296.5032**	**284.0132**	0.0015	0.0023
	Stepwise EM, $\alpha = 0.7$	298.2470	286.1385	0.0016	0.0026
	Stepwise EM, $\alpha = 0.9$	302.3825	291.5596	0.0018	0.0032

Table 2: A summary of the comparison between `SpectralLeader` and stepwise EM with different m and α in real world problems. P1 is YTimes news articles dataset and P2 is PubMed abstracts dataset. The **black font** highlights the best algorithm for a fixed m on one dataset, while the underline highlights the worst algorithm.



(a) (b)

Figure 4: The comparisons between `SpectralLeader` (with reservoir sampling) and stepwise EM on real world problems, where there are 5M words in the streaming data. The x-axis is step n. The y-axis is the average recovery error up to step n.

8 Conclusions

We propose `SpectralLeader`, a novel online learning algorithm for latent variable models. With an instance of bag-of-words model, we prove that `SpectralLeader` converges to a global optimum and derived an $O(\sqrt{n})$ cumulative regret bound for `SpectralLeader`. Experiment results show that `SpectralLeader` is more robust than online EM. In most cases, it performs similar to or better than an optimally-tuned online EM. In the future work, we would like to extend `SpectralLeader` to more complicated latent-variable models, such as HMMs and LDA.

References

[Anandkumar *et al.*, 2012a] Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu.

A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

[Anandkumar *et al.*, 2012b] Animashree Anandkumar, Daniel Hsu, and Sham M Kakade. A method of moments for mixture models and hidden markov models. In *Conference on Learning Theory*, pages 33–1, 2012.

[Anandkumar *et al.*, 2014] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[Cappé and Moulines, 2009] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[Chaganty and Liang, 2013] Arun T Chaganty and Percy Liang. Spectral experts for estimating mixtures of linear regressions. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1040–1048, 2013.

[Ge *et al.*, 2015] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[Hoffman *et al.*, 2010] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[Huang *et al.*, 2015] Furong Huang, UN Niranjan, Mohammad Umar Hakeem, and Animashree Anandkumar. Online tensor methods for learning latent variable models. *Journal of Machine Learning Research*, 16:2797–2835, 2015.

[Kar *et al.*, 2014] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pages 694–702, 2014.

[Liang and Klein, 2009] Percy Liang and Dan Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619. Association for Computational Linguistics, 2009.

[Lichman, 2013] M. Lichman. UCI machine learning repository, 2013.

[Neal and Hinton, 1998] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[Nowozin and Lampert, 2011] Sebastian Nowozin and Christoph H Lampert. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

[Rabiner, 1989] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Shalev-Shwartz, 2012] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[Tung and Smola, 2014] Hsiao-Yu Tung and Alexander J Smola. Spectral methods for indian buffet process inference. In *Advances in Neural Information Processing Systems*, pages 1484–1492, 2014.

[Tung *et al.*, 2017] Hsiao-Yu Fish Tung, Chao-Yuan Wu, Manzil Zaheer, and Alexander J Smola. Spectral methods for nonparametric models. *arXiv preprint arXiv:1704.00003*, 2017.

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

[Wallach, 2006] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.