Novel Document Detection using Online L1-Dictionary Learning

Shiva Prasad Kasiviswanathan
IBM T.J. Watson Research
kasivisw@gmail.com

Huahua Wang
University of Minnesota
huwang@cs.umn.edu

Arindam Banerjee
University of Minnesota
banerjee@cs.umn.edu

Prem Melville
IBM T.J. Watson Research
pmelvil@us.ibm.com

ABSTRACT

Given their pervasive use, social media, such as Twitter, have become a leading source of breaking news. A key task in the automated identification of such news is the detection of novel documents from a voluminous stream of text documents in a robust and scalable manner. In this paper, we introduce an approach for novel document detection based on online dictionary learning. Unlike traditional dictionary learning, which uses squared loss, the proposed formulation uses ℓ_1-loss for the reconstruction error. The online ℓ_1-dictionary learning problem is efficiently solved using the alternating directions method, and we establish a $O(\sqrt{T})$ regret bound under suitable assumptions. Empirical results on news-stream and Twitter data, show more than an order of magnitude speedup, without much loss in quality of results.

1. INTRODUCTION

The high volume and velocity of social media, such as blogs and Twitter, have propelled them to the forefront as sources of breaking news. On Twitter, it is possible to find the latest updates on diverse topics, from natural disasters to celebrity deaths; and identifying such emerging topics has many practical applications, such as in marketing, disease control, and national security [15]. The key challenge in automatic detection of breaking news, is being able to detect novel documents in a stream of text; where a document is considered novel if it is "unlike" documents seen in the past. Recently, this has been made possible by *dictionary learning*, which has emerged as a powerful data representation framework. In dictionary learning each data point $\mathbf{y}$ is represented as a sparse linear combination $A\mathbf{x}$ of dictionary atoms, where A is the dictionary and $\mathbf{x}$ is a sparse vector [1, 13]. A dictionary learning approach can be easily converted into a novel document detection method: let A be a dictionary representing all documents till time $t-1$, for a new data document $\mathbf{y}$ arriving at time t, if one does not find a sparse combination $\mathbf{x}$ of the dictionary atoms, and the best reconstruction $A\mathbf{x}$ yields a large loss, then $\mathbf{y}$ clearly is not well represented by the dictionary A, and is hence novel compared to documents in the past. At the end of timestep t, the dictionary is updated to represent all the documents till time t.

Kasiviswanathan *et al.* [11] presented such a (batch) dictionary learning approach for detecting novel documents/topics. They used an ℓ_1-penalty on the reconstruction error (instead of squared loss commonly used in the dictionary learning literature) as the ℓ_1-penalty has been found to be more effective for text analysis (see Section 3). They also showed this approach outperforms other techniques, such as a nearest-neighbor approach popular in the related area of *First Story Detection* [17]. We build upon this work, by proposing an efficient algorithm for online dictionary learning with ℓ_1-penalty. Our online dictionary learning algorithm is based on the online alternating directions method introduced by Wang *et al.* [21] to solve online composite optimization problems with additional linear equality constraints. Traditional online convex optimization methods such as [27, 9, 6, 7, 24] require explicit computation of the subgradient making them computationally expensive to be applied in our high volume text setting, whereas in our algorithm the subgradients are computed implicitly. The algorithm has simple closed form updates for all steps yielding a fast and scalable algorithm for updating the dictionary. Under suitable assumptions (to cope with the non-convexity of the dictionary learning problem), we establish an $O(\sqrt{T})$ regret bound for the objective, matching the regret bounds of existing methods [27, 6, 7, 24]. Using this online algorithm for ℓ_1-dictionary learning, we obtain an online algorithm for novel document detection, which we empirically validate on traditional news-streams as well as streaming data from Twitter. Experimental results show a substantial speedup over the batch ℓ_1-dictionary learning based approach of Kasiviswanathan *et al.* [11], without a loss of performance in detecting novel documents.

2. PRELIMINARIES

Notation. Vectors are always column vectors and are denoted by boldface letters. For a matrix Z its norm, $\|Z\|_1 = \sum_{i,j} |z_{ij}|$ and $\|Z\|_F^2 = \sum_{ij} z_{ij}^2$. For arbitrary real matrices the standard inner product is defined as $\langle Y, Z\rangle = \mathrm{Tr}(Y^\top Z)$. We use $\Psi_{\max}(Z)$ to denote the largest eigenvalue of $Z^\top Z$. For a scalar $r \in \mathbb{R}$, let $\mathrm{soft}(r,T) = \mathrm{sign}(r) \cdot \max\{|r| - T, 0\}$. The operators sign and soft are extended to a matrix by applying it to every entry in the matrix. $\mathbf{0}_{m\times n}$ denotes a matrix of all zeros of size $m \times n$ and the subscript is omitted when the dimension of the represented matrix is clear from the context.

Dictionary Learning Background. *Dictionary learning* is the problem of estimating a collection of basis vectors over which a given data collection can be accurately reconstructed, often with sparse encodings. It falls into a general category of techniques known as *matrix factorization*. Classic dictionary learning techniques for sparse representation (see [1, 16, 13] and references therein) consider a finite training set of signals $P = [\mathbf{p}_1, \dots, \mathbf{p}_n] \in \mathbb{R}^{m \times n}$ and optimize the empirical cost function which is defined as $f(A) = \sum_{i=1}^{n} l(\mathbf{p}_i, A)$, where $l(\cdot, \cdot)$ is a loss function such that $l(\mathbf{p}_i, A)$ should be small if A is "good" at representing the signal $\mathbf{p}_i$ in a sparse fashion. Here, $A \in \mathbb{R}^{m \times k}$ is referred to as the *dictionary*. In this paper, we use a ℓ_1-loss function with an ℓ_1-regularization term, and our

$$l(\mathbf{p}_i, A) = \min_{\mathbf{x}} \ \|\mathbf{p}_i - A\mathbf{x}\|_1 + \lambda \|\mathbf{x}\|_1,$$

where λ is the regularization parameter. We define the problem of dictionary learning as that of minimizing the empirical cost $f(A)$. In other words, the dictionary learning is the following optimization problem

$$\min_{A} f(A) = f(A, X)$$
$$\stackrel{\text{def}}{=} \min_{A,X} \sum_{i=1}^{n} l(\mathbf{p}_i, A) = \min_{A,X} \|P - AX\|_1 + \lambda \|X\|_1. \quad (1)$$

For maintaining interpretability of the results, we would additionally require that the A and X matrices be non-negative. We also add scaling constraints on A. The optimization problem (1) is in general non-convex. But if one of the variables, either A or X is known, the objective function with respect to the other variable becomes a convex function (in fact, a linear function).

3. NOVEL DOCUMENT DETECTION USING DICTIONARY LEARNING

In this section, we describe the problem of novel document detection and explain how dictionary learning could be used to tackle this problem. Our problem setup is similar to [11].

Novel Document Detection Task. We assume documents arrive in streams. Let $\{P_t \ : \ P_t \in \mathbb{R}^{m_t \times n_t}, t = 1, 2, 3, \dots\}$ denote a sequence of streaming matrices where each column of P_t represents a document arriving at time t. Here, P_t represents the term-document matrix observed at time t. Each document is represented is some conventional vector space model such as TF-IDF [14]. The t could be at any granularity, e.g., it could be the day that the document arrives. We use n_t to represent the number of documents arriving at time t. We normalize P_t such that each column (document) in P_t has a unit ℓ_1-norm. For simplicity in exposition, we will assume that $m_t = m$ for all t.[1] We use the notation $P_{[t]}$ to denote the term-document matrix obtained by vertically concatenating the matrices $P_1, \dots, P_t$, i.e., $P_{[t]} = [P_1|P_2|\dots|P_t]$. Let N_t be the number of documents arriving at time $\leq t$, then $P_{[t]} \in \mathbb{R}^{m \times N_t}$. Under this setup, the *goal* of novel document detection is to identify documents in P_t that are "dissimilar" to the documents in $P_{[t-1]}$.

Sparse Coding to Detect Novel Documents. Let $A_t \in \mathbb{R}^{m \times k}$ represent the dictionary matrix after time $t-1$; where dictionary A_t is a good basis to represent of all the documents in $P_{[t-1]}$. The exact construction of the dictionary is described later. Now, consider a document $\mathbf{y} \in \mathbb{R}^m$ appearing at time t. We say that it admits a *sparse* representation over A_t, if $\mathbf{y}$ could be "well" approximated as a linear combination of few columns from A_t. Modeling a vector with such a sparse decomposition is known as *sparse coding*. In most practical situations it may not be possible to represent $\mathbf{y}$ as $A_t\mathbf{x}$, e.g., if $\mathbf{y}$ has new words which are absent in A_t. In such cases, one could represent $\mathbf{y} = A_t\mathbf{x} + \mathbf{e}$ where $\mathbf{e}$ is an unknown noise vector. We consider the following sparse coding formulation

$$l(\mathbf{y}, A_t) = \min_{\mathbf{x} \geq \mathbf{0}} \ \|\mathbf{y} - A_t\mathbf{x}\|_1 + \lambda \|\mathbf{x}\|_1. \quad (2)$$

The formulation (2) naturally takes into account both the reconstruction error (with the $\|\mathbf{y} - A_t\mathbf{x}\|_1$ term) and the complexity of the sparse decomposition (with the $\|\mathbf{x}\|_1$ term). It is quite easy to transform (2) into a linear program. Hence, it can be solved using a variety of methods. In our experiments, we use the alternating directions method of multipliers (ADMM) [3] to solve (2). ADMM has recently gathered significant attention in the machine learning community due to its wide applicability to a range of learning problems with complex objective functions [3].

We can use sparse coding to detect novel documents as follows. For each document $\mathbf{y}$ arriving at time t, we do the following. First, we solve (2) to check whether $\mathbf{y}$ could be well approximated as a sparse linear combination of the atoms of A_t. If the objective value $l(\mathbf{y}, A_t)$ is "big" then we mark the document as *novel*, otherwise we mark the document as *non-novel*. Since, we have normalized all documents in P_t to unit ℓ_1-length, the objective values are in the same scale.

Choice of the Error Function. A very common choice of reconstruction error is the ℓ_2-penalty. In fact, in the presence of isotopic Gaussian noise the ℓ_2-penalty on $\mathbf{e} = \mathbf{y} - A_t\mathbf{x}$ gives the best approximation of $\mathbf{x}$ [23, 25]. However, for text documents, the noise vector $\mathbf{e}$ rarely satisfies the Gaussian assumption, as some of its coefficients contain large, impulsive values. For example, in fields such as politics and sports, a certain term may become suddenly dominant in a discussion [11]. In presence of non-smooth noise the ℓ_2-penalty on the reconstruction error may give an extremely bad approximation of $\mathbf{x}$ [25]. Our ℓ_1-formulation is inspired by recent results [23, 26, 22] showing that imposing an ℓ_1-penalty gives a more robust and better approximation of $\mathbf{x}$ when the data may contain large, impulsive noise. Additionally, [26] have shown that even without impulsive noise the ℓ_1-penalty does not harm the solution quality as long as the data does not contain a large amount of Gaussian noise.

3.1 Batch Alg. for Novel Document Detection

In this section, we describe a batch algorithm (slightly modified from [11]) for detecting novel documents. The Algorithm BATCH alternates between a novel document detection and a batch dictionary learning step. Let $\mathcal{A}$ be the convex set of matrices defined as

$$\mathcal{A} = \{A \in \mathbb{R}^{m \times k} \ : \ A \geq \mathbf{0}_{m \times k} \ \ \forall j = 1, \dots, k \ , \|A_j\|_1 \leq 1\},$$

where A_j is the jth column in A. We use $\Pi_{\mathcal{A}}$ to denote the projection onto the nearest point in the convex set $\mathcal{A}$.

Batch Dictionary Learning. We now describe the batch dictionary learning step. Firstly, we require the A's and X's to be non-negative for interpretability purposes. To prevent the dictionary from having arbitrarily large values (which could lead to small values for X's), we also require that each column of the dictionary have a ℓ_1-norm less than or equal to 1. Adding this constraint, the

[1] As new documents come in and new terms are identified, we expand the vocabulary and zero-pad the previous matrices so that at the current time t, all previous and current documents have a representation over the same vocabulary space.

Algorithm 1 : BATCH

Input: $P_{[t-1]} \in \mathbb{R}^{m \times N_{t-1}}$, $P_t = [\mathbf{p}_1, \dots, \mathbf{p}_{n_t}] \in \mathbb{R}^{m \times n_t}$, $A_t \in \mathbb{R}^{m \times k}$, $\lambda \geq 0, \zeta \geq 0$
Novel Document Detection Step:
for $j = 1$ **to** n_t **do**
  Solve: $\mathbf{x}_j = \text{argmin}_{\mathbf{x} \geq 0} \|\mathbf{p}_j - A_t \mathbf{x}\|_1 + \lambda \|\mathbf{x}\|_1$
  if $\|\mathbf{p}_j - A_t \mathbf{x}_j\|_1 + \lambda \|\mathbf{x}_j\|_1 > \zeta$
    Mark $\mathbf{p}_j$ as novel
Batch Dictionary Learning Step:
Set $P_{[t]} \leftarrow [P_{[t-1]} \,|\, \mathbf{p}_1, \dots, \mathbf{p}_{n_t}]$
Solve: $[A_{t+1}, X_{[t]}] = \text{argmin}_{A \in \mathcal{A}, X \geq \mathbf{0}} \|P_{[t]} - AX\|_1 + \lambda \|X\|_1$

dictionary learning step becomes[2]

$$[A_{t+1}, X_{[t]}] = \text{argmin}_{A \in \mathcal{A}, X \geq \mathbf{0}} \|P_{[t]} - AX\|_1 + \lambda \|X\|_1. \quad (3)$$

The above optimization problem could be transformed into a linear program.

Even though conceptually simple, Algorithm BATCH is computationally inefficient. The bottleneck comes in the dictionary learning step. As t increases, so does the size of $P_{[t]}$, so solving (3) becomes prohibitive even with efficient optimization techniques. To achieve computational efficiency, in [11], the authors solved an approximation of (3) where in the dictionary learning step they only update the A's and not the X's.[3] This leads to faster running times, but because of the approximation, the quality of the dictionary degrades over time and the performance of the algorithm decreases. In this paper, we propose an online learning algorithm for (3) and show that this online algorithm is both computationally efficient and generates good quality dictionaries under reasonable assumptions.

4. ONLINE L1-DICTIONARY UPDATE

In this section, we introduce the online ℓ_1-dictionary learning problem and propose an efficient algorithm for it. An online algorithm performs well if its *regret* is sublinear in time T, since this implies that "on the average" the algorithm performs as well as the best fixed strategy in hindsight [19]. Now consider the ℓ_1-dictionary learning problem defined in (3). Since this problem is jointly non-convex in (A, X) it is not possible to obtain an efficient online algorithm with sublinear regret without making any assumptions on either the dictionary (A) or the sparse codes (X) (because an online algorithm with sublinear regret would imply an efficient algorithm for solving (3) in the batch case). Therefore, we focus on obtaining regret bounds for the dictionary update, assuming that the at each timestep the sparse codes given to the batch and online algorithms are the same. This motivates the following problem.

DEFINITION 4.1 (ONLINE ℓ_1-DICTIONARY PROBLEM). *At time t, the online algorithm picks $\hat{A}_{t+1} \geq \mathbf{0}, \hat{A}_{t+1} \in \mathcal{A}$, based on $(P_t, X_t, \hat{A}_t)$. Then, nature (adversary) reveals (P_{t+1}, X_{t+1}) with $P_{t+1} \in \mathbb{R}^{m \times n}$ and $X_{t+1} \in \mathbb{R}^{m \times k}$. The problem is to pick the $\hat{A}_{t+1}$ sequence such that the following regret function is minimized*[4]

$$R(T) = \sum_{t=1}^{T} \|P_t - \hat{A}_t X_t\|_1 - \min_{A \geq \mathbf{0}, A \in \mathcal{A}} \sum_{t=1}^{T} \|P_t - AX_t\|_1.$$

Regret is the de-facto standard in measuring performance of online learning algorithms. Intuitively, an algorithm performs well if its regret is sublinear in T, since this implies that "on the average" the algorithm performs as well as the best fixed strategy in hindsight [27, 7].

4.1 Online L1-Dictionary Algorithm

In this section, we design an algorithm for online ℓ_1-dictionary learning problem. which we call Online Inexact ADMM (OIADMM), and bound its regret. Firstly note that because of the non-smooth ℓ_1-norms involved it is computationally expensive to apply standard online learning algorithms like online (stochastic) gradient descent [27, 9], COMID [7], FOBOS [6], and RDA [24], as they require computing a costly subgradient at every iteration. We propose a variant of the online alternating direction method, which was first proposed by Wang *et al.* [21]. In the proposed method, we first perform a simple variable substitution by introducing an equality constraint. The update for each variable has a closed-form solution without the need of estimating the subgradients explicitly. Furthermore, each update amounts to a simple element-wise operation and can be done in parallel resulting in a highly scalable algorithm.

The operation of the algorithm is simple. At each time t it uses the following minimization problem to get $\hat{A}_{t+1}$:

$$\min_{A \geq \mathbf{0}, A \in \mathcal{A}} \|P_t - AX_t\|_1.$$

We can rewrite this above minimization problem as:

$$\min_{A \geq \mathbf{0}, A \in \mathcal{A}, \Gamma} \|\Gamma\|_1 \quad \text{s.t.} \quad P_t - AX_t = \Gamma. \qquad (4)$$

The augmented Lagrangian of (4) is:

$$\mathcal{L}(A, \Gamma, \Delta_t) = \min_{A \geq \mathbf{0}, A \in \mathcal{A}, \Gamma} \|\Gamma\|_1 + \langle \Delta_t, P_t - AX_t - \Gamma \rangle + \frac{\beta_t}{2} \|P_t - AX_t - \Gamma\|_F^2, \quad (5)$$

where $\Delta_t \in \mathbb{R}^{m \times n}$ is a multiplier and $\beta_t > 0$ is a penalty parameter.

OIADMM is summarized in Algorithm 2. It is based on updating the variables Γ, A, and Δ_t. Instead of solving (4) completely, at each time t it only runs one step ADMM update of the variables. OIADMM gets as input P_t, X_t, $\hat{A}_t$, and Δ_t. It outputs $\hat{A}_{t+1}$ and Δ_{t+1}. All the updates in OIADMM have simple closed-form.

Let $\widetilde{\Gamma}_t = P_t - \hat{A}_t X_t$. First for a fixed $\hat{A}_t$ and Δ_t, Γ that minimizes (5) could be obtained by minimizing

$$\underset{\Gamma}{\text{argmin}} \|\Gamma\|_1 + \langle \Delta_t, \widetilde{\Gamma}_t - \Gamma \rangle + \frac{\beta_t}{2} \|\widetilde{\Gamma}_t - \Gamma\|_F^2. \qquad (6)$$

The Γ that minimizes (6) is Γ_{t+1}. Now using fixed Γ_{t+1} and Δ_t, a simple manipulation shows that we can obtain the A that minimizes (5) by solving

$$\min_{A \geq \mathbf{0}, A \in \mathcal{A}} \frac{\beta_t}{2} \left\| P_t - AX_t - \Gamma_{t+1} + \frac{\Delta_t}{\beta_t} \right\|_F^2. \qquad (7)$$

[2]In our algorithms, it is quite straightforward to replace the condition $A \in \mathcal{A}$ by some other condition $A \in \mathcal{C}$, where $\mathcal{C}$ is some closed non-empty convex set.

[3]In particular, define $\widetilde{X}_{[t]} = [\widetilde{X}_{[t-1]} \,|\, \mathbf{x}_1, \dots, \mathbf{x}_{n_t}]$ where $\mathbf{x}_j$'s are coming from the novel document detection step at time t and $\widetilde{X}_{[t-1]}$ comes from timestep $t-1$. In [11], the dictionary learning step is $A_{t+1} = \text{argmin}_{A \in \mathcal{A}} \|P_{[t]} - A\widetilde{X}_{[t]}\|_1$.

[4]For ease of presentation and analysis, we will assume that m and n don't vary with time. One could allow for changing m and n by carefully adjusting the size of the matrices by zero-padding.

Algorithm 2 : OIADMM

Input: $P_t \in \mathbb{R}^{m\times n}$, $\hat{A}_t \in \mathbb{R}^{m\times k}$, $\Delta_t \in \mathbb{R}^{m\times n}$, $\hat{X}_t \in \mathbb{R}^{k\times n}$, $\beta_t \geq 0, \tau_t \geq 0$

$\widetilde{\Gamma}_t \longleftarrow P_t - \hat{A}_t\hat{X}_t$

$\Gamma_{t+1} = \mathrm{argmin}_\Gamma \ \|\Gamma\|_1 + \langle \Delta_t, \widetilde{\Gamma}_t - \Gamma\rangle + (\beta_t/2)\|\widetilde{\Gamma}_t - \Gamma\|_F^2$ $\quad (\Rightarrow \Gamma_{t+1} = \mathrm{soft}(\widetilde{\Gamma}_t + \Delta_t/\beta_t, 1/\beta_t))$

$G_{t+1} \longleftarrow -(\Delta_t/\beta_t + \widetilde{\Gamma}_t - \Gamma_{t+1})\hat{X}_t^\top$

$\hat{A}_{t+1} = \mathrm{argmin}_{A\in\mathcal{A}} \ \beta_t(\langle G_{t+1}, A - \hat{A}_t\rangle + (1/2\tau_t)\|A - \hat{A}_t\|_F^2 \ )$ $\quad (\Rightarrow \hat{A}_{t+1} = \Pi_\mathcal{A}(\max\{0, \hat{A}_t - \tau_t G_{t+1}\}))$

$\Delta_{t+1} = \Delta_t + \beta_t(P_t - \hat{A}_{t+1}\hat{X}_t - \Gamma_{t+1})$

Return $\hat{A}_{t+1}$ and Δ_{t+1}

Instead of solving (7) exactly, we approximate it by

$$\min_{A\geq \mathbf{0}, A\in\mathcal{A}} \beta_t(\langle G_{t+1}, A - \hat{A}_t\rangle + \frac{1}{2\tau_t}\|A - \hat{A}_t\|_F^2), \tag{8}$$

where $\tau_t > 0$ is a proximal parameter and G_{t+1} is the gradient of $\|P_t - AX_t - \Gamma_{t+1} + \Delta_t/\beta_t\|_F^2$ at $A = \hat{A}_t$. The above approach belongs to the class of proximal gradient methods in optimization [20, 26]. The A that minimizes (8) is $\hat{A}_{t+1}$. We use $\Pi_\mathcal{A}$ to denote the projection onto the convex set $\mathcal{A}$. Now that we have obtained Γ_{t+1} and $\hat{A}_{t+1}$, we update Δ as $\Delta_{t+1} = \Delta_t + \beta_t(P_t - \hat{A}_{t+1}X_t - \Gamma_{t+1})$. OIADMM allows the temporary violation of equality constraint in (4), but satisfies the constraint on average in the long run. More formally, at each time t it could happen that $\hat{A}_{t+1}$ and Γ_{t+1} produced by OIADMM is such that $P_t - \hat{A}_{t+1}X_t \neq \Gamma_{t+1}$. However, we show that the algorithm satisfies a long term constraint such that $\sum_{t=1}^T \|\Gamma_{t+1} - P_t + \hat{A}_{t+1}X_t\|_2^2$ is sublinear in T (Theorem 4.7).

Analysis of OIADMM**.** Let A_{opt} be the optimum solution to $\min_{A\geq\mathbf{0},A\in\mathcal{A}} \sum_{t=1}^T \|P_t - AX_t\|_1$. Define, $\Gamma^\star_{\mathrm{opt}} = P_t - A_{\mathrm{opt}}X_t$. The regret of the OIADMM is

$$R(T) = \sum_{t=1}^T \|\widetilde{\Gamma}_t\|_1 - \|\Gamma_{\mathrm{opt}}\|_1.$$

Let $\widehat{\Gamma}_t = P_t - \hat{A}_{t+1}X_t$. For any, $A^\star \geq \mathbf{0}, A^\star \in \mathcal{A}$, let $\Gamma_t^\star = P_t - A^\star X_t$. The lemmas below hold for any $A^\star \geq \mathbf{0}, A^\star \in \mathcal{A}$ so in particular it holds for $A^\star = A_{\mathrm{opt}}$. We split the proof into three technical lemmas. We first upper bound $\langle \Delta_t, \widehat{\Gamma}_t - \Gamma_t^\star\rangle$ (Lemma 4.3), and use it to bound $\|\Gamma_{t+1}\|_1 - \|\Gamma_t^\star\|_1$ (Lemma 4.4). In the proof of Lemma 4.5, we bound $\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_{t+1}\|_1$ and this when added to the bound on $\|\Gamma_{t+1}\|_1 - \|\Gamma_t^\star\|_1$ gives a bound on one-step loss $\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1$. We would use the following well-known lemma in the proof.

LEMMA 4.2. *For matrices $M_1, M_2, M_3, M_4 \in \mathbb{R}^{m\times n}$, we have the following*

$$2\langle M_1 - M_2, M_3 - M_4\rangle = \|M_1 - M_4\|_F^2 + \|M_2 - M_3\|_F^2 - \|M_1 - M_3\|_F^2 - \|M_2 - M_4\|_F^2.$$

LEMMA 4.3. *Let $\{\Gamma_t, \hat{A}_t, \Delta_t\}$ be the sequences generated by the* OIADMM *procedure. For any $A^\star \geq \mathbf{0}$ and $A^\star \in \mathcal{A}$, we have*

$$\langle \Delta_t, \widehat{\Gamma}_t - \Gamma_t^\star\rangle \leq \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) + \frac{\beta_t}{2}\left(\|\Gamma_t^\star - \Gamma_{t+1}\|_F^2 - \|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 - \|\Gamma_t^\star - \widetilde{\Gamma}_t\|_F^2\right) - \frac{\beta_t}{2}(\frac{1}{\tau_t} - \Psi_{\max}(X_t))\|\hat{A}_{t+1} - \hat{A}_t\|_F^2. \tag{9}$$

PROOF. For any $A^\star \geq \mathbf{0}$ and $A^\star \in \mathcal{A}$, (8) is equivalent to the following variational inequality [18]:

$$\beta_t\langle G_{t+1} + \frac{1}{\tau_t}(\hat{A}_{t+1} - \hat{A}_t), A^\star - \hat{A}_{t+1}\rangle \geq 0. \tag{10}$$

Using $\widehat{\Gamma}_t = P_t - \hat{A}_{t+1}X_t$ and substituting for G_{t+1}, we have

$$\begin{aligned}
&\beta_t\langle G_{t+1}, A^\star - \hat{A}_{t+1}\rangle \qquad (11)\\
&= -\beta_t\langle(\Delta_t/\beta_t + \widetilde{\Gamma}_t - \Gamma_{t+1})X_t^\top, A^\star - \hat{A}_{t+1}\rangle\\
&= \beta_t\langle \Delta_t/\beta_t + \widetilde{\Gamma}_t - \Gamma_{t+1}, \hat{A}_{t+1}X_t - A^\star X_t\rangle\\
&= \beta_t\langle \Delta_t/\beta_t + \widetilde{\Gamma}_t - \Gamma_{t+1}, P_t - A^\star X_t - (P_t - \hat{A}_{t+1}X_t)\rangle\\
&= \langle \Delta_t, \Gamma_t^\star - \widehat{\Gamma}_t\rangle + \beta_t\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_t^\star - \widehat{\Gamma}_t\rangle. \qquad (12)
\end{aligned}$$

Substituting (11) into (10) and rearranging the terms yield

$$\langle \Delta_t, \widehat{\Gamma}_t - \Gamma_t^\star\rangle \leq \beta_t\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_t^\star - \widehat{\Gamma}_t\rangle + \frac{\beta_t}{\tau_t}\langle \hat{A}_{t+1} - \hat{A}_t, A^\star - \hat{A}_{t+1}\rangle. \tag{13}$$

By using Lemma 4.2, the first term on the right side can be rewritten as

$$\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_t^\star - \widehat{\Gamma}_t\rangle = \frac{1}{2}\big(\|\widetilde{\Gamma}_t - \widehat{\Gamma}_t\|_F^2 + \|\Gamma_t^\star - \Gamma_{t+1}\|_F^2 - \|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 - \|\Gamma_t^\star - \widetilde{\Gamma}_t\|_F^2\big). \tag{14}$$

Substituting the definitions of $\widehat{\Gamma}_t$ and $\widetilde{\Gamma}_t$, we have

$$\|\widetilde{\Gamma}_t - \widehat{\Gamma}_t\|_F^2 = \|P_t - \hat{A}_tX_t - (P_t - \hat{A}_{t+1}X_t)\|_F^2 = \|(\hat{A}_{t+1} - \hat{A}_t)X_t\|_F^2 \leq \Psi_{\max}(X_t)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2, \tag{15}$$

Remember that $\Psi_{\max}(X_t)$ is the maximum eigenvalue of $X^\top X$. Using Lemma 4.2, we get that the second term in the right hand side of (13) is equivalent to

$$\langle \hat{A}_{t+1} - \hat{A}_t, A^\star - \hat{A}_{t+1}\rangle = \frac{1}{2}\big(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2 - \|\hat{A}_{t+1} - \hat{A}_t\|_F^2\big). \tag{16}$$

Combining results in (13), (14), (15), and (16), we get the desired bound. $\square$

LEMMA 4.4. *Let $\{\Gamma_t, \hat{A}_t, \Delta_t\}$ be the sequences generated by the OIADMM procedure. For any $A^\star \geq \mathbf{0}$ and $A^\star \in \mathcal{A}$, we have*

$$\|\Gamma_{t+1}\|_1 - \|\Gamma_t^\star\|_1 \leq \frac{1}{2\beta_t}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) + \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) - \frac{\beta_t}{2}(\frac{1}{\tau_t} - \Psi_{\max}(X_t))\|\hat{A}_{t+1} - \hat{A}_t\|_F^2 - \frac{\beta_t}{2}\|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2. \tag{17}$$

PROOF. Let $\partial\|\Gamma_{t+1}\|_1$ denote the subgradient of $\|\Gamma_{t+1}\|_1$. Now Γ_{t+1} is a minimizer of (6). Therefore,

$$\mathbf{0} \in \partial\|\Gamma_{t+1}\|_1 - \Delta_t - \beta_t(\widetilde{\Gamma}_t - \Gamma_{t+1}).$$

Rearranging the terms gives $\Delta_t + \beta_t(\widetilde{\Gamma}_t - \Gamma_{t+1}) \in \partial\|\Gamma_{t+1}\|_1$. Since $\|\Gamma_{t+1}\|_1$ is a convex function, we have

$$\begin{aligned}&\|\Gamma_{t+1}\|_1 - \|\Gamma_t^\star\|_1 \leq \langle \Delta_t + \beta_t(\widetilde{\Gamma}_t - \Gamma_{t+1}), \Gamma_{t+1} - \Gamma_t^\star\rangle \\ &\leq \langle \Delta_t, \Gamma_{t+1} - \widehat{\Gamma}_t\rangle + \langle \Delta_t, \widehat{\Gamma}_t - \Gamma_t^\star\rangle + \beta_t\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_{t+1} - \Gamma_t^\star\rangle.\end{aligned} \tag{18}$$

Using Lemma 4.2, the last term can be rewritten as

$$\begin{aligned}&\beta_t\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_{t+1} - \Gamma_t^\star\rangle \\ &= \frac{\beta_t}{2}\left(\|\Gamma_t^\star - \widetilde{\Gamma}_t\|_F^2 - \|\Gamma_t^\star - \Gamma_{t+1}\|_F^2 - \|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2\right)\end{aligned} \tag{19}$$

Combining (9) with (19) gives

$$\begin{aligned}&\langle \Delta_t, \widehat{\Gamma}_t - \Gamma_t^\star\rangle + \beta_t\langle \widetilde{\Gamma}_t - \Gamma_{t+1}, \Gamma_{t+1} - \Gamma_t^\star\rangle \\ &\quad\leq \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) \\ &\quad\quad - \frac{\beta_t}{2}\left(\frac{1}{\tau_t} - \Psi_{\max}(X_t)\right)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2 \\ &\quad\quad - \frac{\beta_t}{2}\left(\|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2 - \|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2\right).\end{aligned} \tag{20}$$

Since $\Gamma_{t+1} - \widehat{\Gamma}_t = (\Delta_t - \Delta_{t+1})/\beta_t$, we have

$$\begin{aligned}&\langle \Delta_t, \Gamma_{t+1} - \widehat{\Gamma}_t\rangle - \frac{\beta_t}{2}\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 \\ &= \frac{1}{2\beta_t}\left(2\langle \Delta_t, \Delta_t - \Delta_{t+1}\rangle - \|\Delta_t - \Delta_{t+1}\|_F^2\right) \\ &= \frac{1}{2\beta_t}\left(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2\right).\end{aligned} \tag{21}$$

Plugging (20) and (21) into (18) yields the result. □

LEMMA 4.5. *Let $\{\Gamma_t, \hat{A}_t, \Delta_t\}$ be the sequences generated by the OIADMM procedure. If τ_t satisfies $\frac{1}{\tau_t} \geq 2\Psi_{\max}(X_t)$. Then*

$$\begin{aligned}\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1 &\leq \frac{1}{2\beta_t}\|\Lambda_t\|_F^2 + \frac{1}{2\beta_t}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) \\ &\quad + \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right),\end{aligned} \tag{22}$$

where $\Lambda_t \in \partial\|\widetilde{\Gamma}_t\|_1$.

PROOF. Let $\Lambda_t \in \partial\|\widetilde{\Gamma}_t\|_1$. Therefore,

$$\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_{t+1}\|_1 \leq \langle \Lambda_t, \widetilde{\Gamma}_t - \Gamma_{t+1}\rangle.$$

Now,

$$\begin{aligned}\langle \Lambda_t, \widetilde{\Gamma}_t - \Gamma_{t+1}\rangle &= \langle \Lambda_t/\sqrt{\beta_t}, \sqrt{\beta_t}(\widetilde{\Gamma}_t - \Gamma_{t+1})\rangle \\ &\leq \frac{1}{2\beta_t}\|\Lambda_t\|_F^2 + \frac{\beta_t}{2}\|\widetilde{\Gamma}_t - \Gamma_{t+1}\|_F^2\end{aligned}$$

Therefore,

$$\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_{t+1}\|_1 \leq \frac{1}{2\beta_t}\|\Lambda_t\|_F^2 + \frac{\beta_t}{2}\|\widetilde{\Gamma}_t - \Gamma_{t+1}\|_F^2. \tag{23}$$

Adding (23) and (17) together we get

$$\begin{aligned}\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1 &\leq \frac{1}{2\beta_t}\|\Lambda_t\|_F^2 + \frac{1}{2\beta_t}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) \\ &\quad + \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) \\ &\quad - \frac{\beta_t}{2}\left(\frac{1}{\tau_t} - \Psi_{\max}(X_t)\right)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2.\end{aligned}$$

Setting $\frac{1}{\tau_t} \geq 2\Psi_{\max}(X_t)$ means that

$$-\frac{\beta_t}{2}\left(\frac{1}{\tau_t} - \Psi_{\max}(X_t)\right)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2 \leq 0.$$

Therefore,

$$\begin{aligned}\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1 &\leq \frac{1}{2\beta_t}\|\Lambda_t\|_F^2 + \frac{1}{2\beta_t}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) \\ &\quad + \frac{\beta_t}{2\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right),\end{aligned}$$

□

The following regret bound theorem follows by using a canceling telescoping sum on the bound on $\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1$.

THEOREM 4.6. *Let $\{\Gamma_t, \hat{A}_t, \Delta_t\}$ be the sequences generated by the OIADMM procedure and $R(T)$ be defined as above. Assume the following conditions hold: (i) the Frobenius norm of $\partial\|\Gamma_t\|_1$ is upper bounded by Φ, (ii) $\hat{A}_0 = \mathbf{0}$, $\|A^\star\|_F \leq D$, (iii) $\Delta_0 = \mathbf{0}$, and (iv) $1/\tau_t \geq 2\Psi_{\max}(X_t)$. Setting $\beta_t = (\Phi/D)\sqrt{\tau_t T}$, we have*

$$R(T) \leq \frac{\Phi D\sqrt{T}}{2\sqrt{\tau_t}}.$$

PROOF. Summing (22) over t from 1 to T, we get the following canceling telescoping sum

$$\begin{aligned}&\sum_{t=1}^{T}\|\widetilde{\Gamma}_t\|_1 - \|\Gamma_t^\star\|_1 \leq \frac{1}{2\beta_t}\sum_{t=1}^{T}\|\Lambda_t\|_F^2 \\ &+ \frac{1}{2\beta_t}(\|\Delta_0\|_F^2 - \|\Delta_T\|_F^2) + \frac{\beta_t}{2\tau_t}(\|A^\star - \hat{A}_0\|_F^2 - \|A^\star - \hat{A}_T\|_F^2) \\ &\leq \frac{\Phi^2 T}{2\beta_t} + \frac{D^2\beta_t}{2\tau_t}.\end{aligned}$$

Setting $\beta_t = \frac{\Phi}{D}\sqrt{\tau_t T}$ yields desired bound. □

OIADMM can violate the equality constraint in (4) at each t. The following result shows that the accumulated loss caused by the violation of equality constraint is sublinear in T. Let Υ be a an upper bound on $\|\Gamma_t^\star\|_1$.

THEOREM 4.7. *Let $\{\Gamma_t, \hat{A}_t, \Delta_t\}$ be the sequences generated by the OIADMM procedure. Letting the assumptions in Theorem 4.6 hold and with $\|\Gamma_t^\star\|_1 \leq \Upsilon$, we have*

$$\sum_{t=1}^{T}\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_2^2 \leq \frac{2D^2}{\tau_t} + \frac{4\Upsilon D}{\Phi\sqrt{\tau_t}}\sqrt{T}.$$

PROOF. Lets look at $\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2$.

$$\begin{aligned}&\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 = \|\Gamma_{t+1} - \widetilde{\Gamma}_t + \widetilde{\Gamma}_t - \widehat{\Gamma}_t\|_F^2 \\ &\leq 2\left(\|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2 + \|\widetilde{\Gamma}_t - \widehat{\Gamma}_t\|_F^2\right) \\ &\leq 2\left(\|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2 + \Psi_{\max}(X_t)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2\right).\end{aligned} \tag{24}$$

For the first inequality, we used the simple fact that for any two matrices M_1 and M_2 $\|M_1 - M_2\|_F^2 \leq 2(\|M_1\|_F^2 + \|M_2\|_F^2)$. The second inequality is because of (15). Firstly, since $\|\Gamma_{t+1}\|_1 \geq 0$

$$\|\Gamma_{t+1}\|_1 - \|\Gamma_t^\star\|_1 \geq -\|\Gamma_t^\star\|_1 \geq -\Upsilon.$$

Using this and rearranging terms in (17) gives

$$\|\Gamma_{t+1} - \widetilde{\Gamma}_t\|_F^2 \leq \frac{1}{\beta_t^2}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) + \frac{1}{\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) - (\frac{1}{\tau_t} - \Psi_{\max}(X_t))\|\hat{A}_{t+1} - \hat{A}_t\|_F^2 + \frac{2\Upsilon}{\beta_t},$$

Plugging this into (24) yields

$$\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 \leq \frac{2}{\beta_t^2}(\|\Delta_t\|_F^2 - \|\Delta_{t+1}\|_F^2) + \frac{2}{\tau_t}\left(\|A^\star - \hat{A}_t\|_F^2 - \|A^\star - \hat{A}_{t+1}\|_F^2\right) - 2\left(\frac{1}{\tau_t} - 2\Psi_{\max}(X_t)\right)\|\hat{A}_{t+1} - \hat{A}_t\|_F^2 + \frac{4\Upsilon}{\beta_t}. \quad (25)$$

Letting $\frac{1}{\tau_t} \geq 2\Psi_{\max}(X_t)$ and summing over t from 1 to T, we have

$$\sum_{t=1}^{T}\|\Gamma_{t+1} - \widehat{\Gamma}_t\|_F^2 \leq \frac{2}{\beta_t^2}(\|\Delta_0\|_F^2 - \|\Delta_T\|_F^2) + \frac{2}{\tau_t}\left(\|A^\star - \hat{A}_0\|_F^2 - \|A^\star - \hat{A}_T\|_F^2\right) + \frac{4\Upsilon T}{\beta_t} \leq \frac{2D^2}{\tau_t} + \frac{4\Upsilon T}{\beta_t}. \quad (26)$$

Setting $\beta_t = \frac{\Phi}{D}\sqrt{\tau_t T}$ yields the desired bound. □

Efficiency and Extensions. The projection onto $\mathcal{A}$ can be efficiently done as explained in [5]. The simplest implementation of OIADMM takes $O(mnk)$ time at each t. However, in practice, we can exploit the sparsity in P_t and X_t to make the algorithm run much faster. Another advantage of OIADMM is that it can easily be distributed in the framework introduced by [3]. Also, OIADMM is memory efficient, as at each time t, we only need A_{t-1} and none of the past inputs. It is also quite straightforward to replace the condition $A \in \mathcal{A}$ by $A \in \mathcal{C}$, where $\mathcal{C}$ is some closed non-empty convex set.

5. EXPERIMENTAL RESULTS

In this section, we present experiments to compare and contrast the performance of ℓ_1-batch and ℓ_1-online dictionary learning algorithms for the task of novel document detection.

Implementation of BATCH. In our implementation, we grow the dictionary size by η in each timestep. Growing the dictionary size is essential for the batch algorithm because as t increases the number of columns of $P_{[t]}$ also increases, and therefore, a larger dictionary is required to compactly represent all the documents in $P_{[t]}$. For solving (3), we use alternative minimization over the variables. We use the ADMM technique to solve the optimization problems arising in the sparse coding and dictionary learning steps.

Online Algorithm for Novel Document Detection. Algorithm ONLINE[5] uses the same novel document detection step as Algorithm BATCH, but dictionary learning is done using OIADMM.

Notice that the sparse coding matrices generated by Algorithm BATCH, $X_1, \dots, X_t$ (where $X_{[t]} = [X_1|\dots|X_t]$) could be different from

Algorithm 3 : ONLINE

Input: $P_t = [\mathbf{p}_1, \dots, \mathbf{p}_{n_t}] \in \mathbb{R}^{m \times n_t}$, $\hat{A}_t \in \mathbb{R}^{m \times k}$, $\Delta_t \in \mathbb{R}^{m \times n_t}$, $\lambda \geq 0, \zeta \geq 0, \beta \geq 0, \tau \geq 0$
Novel Document Detection Step:
for $j = 1$ **to** n_t **do**
  Solve: $\mathbf{x}_j = \mathrm{argmin}_{\mathbf{x} \geq 0} \|\mathbf{p}_j - \hat{A}_t\mathbf{x}\|_1 + \lambda\|\mathbf{x}\|_1$
  if $\|\mathbf{p}_j - \hat{A}_t\mathbf{x}_j\|_1 + \lambda\|\mathbf{x}_j\|_1 > \zeta$
    Mark $\mathbf{p}_j$ as novel
Online Dictionary Learning Step:
Set $\hat{X}_t \longleftarrow [\mathbf{x}_1, \dots, \mathbf{x}_{n_t}]$
$(\hat{A}_{t+1}, \Delta_{t+1}) \longleftarrow$ OIADMM$(P_t, \hat{A}_t, \Delta_t, \hat{X}_t, \beta, \tau)$[6]

$\hat{X}_1, \dots, \hat{X}_t$. If these sequence of matrices are same, then we have a sublinear regret on the objective function.[7]

Experimental Setup. All reported results are based on a Matlab implementation running on a quad-core 2.33 GHz Intel processor with 32GB RAM. The regularization parameter λ is set to 0.1 which yields reasonable sparsities in our experiments. OIADMM parameters τ_t is set $1/(2\Psi_{\max}(\hat{X}_t))$ (chosen according to Theorem 4.6) and β_t is fixed to 5 (obtained through tuning). The ADMM parameters for sparse coding and batch dictionary learning are set as suggested in [11]. In the batch algorithms, we grow the dictionary sizes by $\eta = 10$ in each timestep. The threshold values ζ are treated as tunable parameters.

5.1 Experiments on News Streams

Our first dataset is drawn from the NIST Topic Detection and Tracking (TDT2) corpus which consists of news stories in the first half of 1998. In our evaluation, we used a set of 9000 documents represented over 19528 terms and distributed into the top 30 TDT2 human-labeled topics over a period of 27 weeks. We introduce the documents in groups. At timestep 0, we introduce the first 1000 documents and these documents are used for initializing the dictionary. We use an alternative minimization procedure over the variables of (1) to initialize the dictionary. In these experiments the size of the initial dictionary $k = 200$. In each subsequent timestep $t \in \{1, \dots, 8\}$, we provide the batch and online algorithms the same set of 1000 documents. In Figure 1, we present novel document detection results for those timesteps where at least one novel document was introduced. Table 1 shows the corresponding AUC numbers.

Comparison of the ℓ_1-online and ℓ_1-batch Algorithms. The ℓ_1-online (Algorithm ONLINE) and ℓ_1-batch algorithms have almost identical performance in terms of detecting novel documents (see Table 1). However, the online algorithm is much more computationally efficient. In Figure 2(a), we compare the running times of these algorithms. As noted earlier, the running time of the batch algorithm goes up as t increases (as it has to optimize over the entire past). However, the running time of the online algorithm is independent of the past and only depends on the number of documents introduced in each timestep (which in this case is always 1000). Therefore, the running time of the online algorithm is almost the same across different timesteps. As expected the run-time gap between the ℓ_1-batch and ℓ_1-online algorithms widen as t increases – in the first timestep ONLINE is 5.4 times faster, and this rapidly increases to a factor of 11.5 in just 7 timesteps.

5.2 Experiments on Twitter

Our second dataset is from an application of monitoring Twitter for Marketing and PR for smartphone and wireless providers. We

[5]In our experiments, the number of documents introduced in each timestep is almost of the same order, and hence there is no need to change the size of the dictionary across timesteps for the Algorithm ONLINE.

[6]Before invoking Algorithm OIADMM we have to zero-pad the matrices in the arguments appropriately.

[7]As noted earlier, we can not do a comparison without making any assumptions.

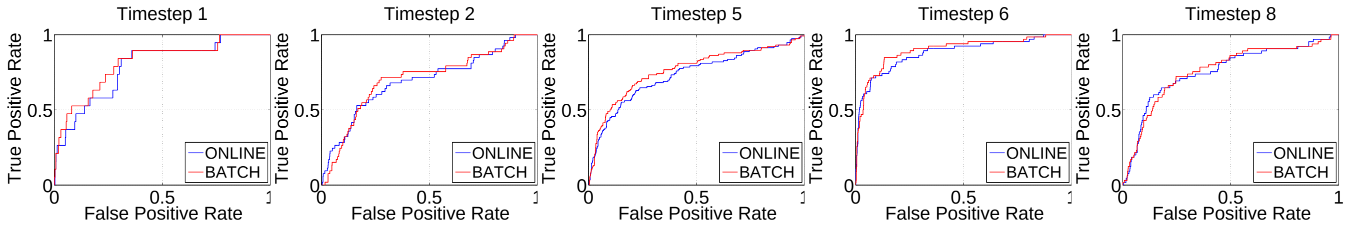

Figure 1: ROC curves for TDT2 for timesteps where novel documents were introduced.

Timestep	*No. of Novel Docs.*	*No. of Nonnovel Docs.*	*AUC ℓ_1-online*	*AUC ℓ_1-batch*
1	19	981	0.791	0.815
2	53	947	0.694	0.704
5	116	884	0.732	0.764
6	66	934	0.881	0.898
8	65	935	0.757	0.760
Avg.			**0.771**	**0.788**

Table 1: AUC Numbers for ROC Plots in Figure 1.

used the Twitter Decahose to collect a 10% sample of all tweets (posts) from Sept 15 to Oct 05, 2011. From this, we filtered the tweets relevant to "Smartphones" using a scheme presented in [4] which utilizes the Wikipedia ontology to do the filtering. Our dataset comprises of 127760 tweets over these 21 days and the vocabulary size is 6237 words. We used the tweets from Sept 15 to 21 (34292 in number) to initialize the dictionaries. Subsequently, at each timestep, we give as input to both the algorithms all the tweets from a given day (for a period of 14 days between Sept 22 to Oct 05). Since this dataset is unlabeled, we do a a qualitative evaluation of the ℓ_1-online algorithm for the novel document detection task. Here, the size of the intial dictionary $k = 100$.

Figure 2(b) shows the running times on the Twitter dataset. At first timestep the online algorithm is already 10.8 times faster, and this speedup escalates to 18.2 by the 14th timestep.

Table 2 below shows a representative set of novel tweets identified by Algorithm ONLINE. In each timestep, instead of thresholding by ζ, we take the top 10% of tweets measured in terms of the sparse coding objective value and run a dictionary-based clustering, described in [11], on it. Further post-processing is done to discard clusters without much support and to pick a representative tweet for each cluster. Using this completely automated process, we are able to detect breaking news and trending relevant to the smartphone market, such as AT&T throttling data bandwidth, launch of IPhone 4S, and the death of Steve Jobs.

6. CONCLUSION

The main contribution of this paper is a new online ℓ_1-dictionary learning algorithm, based on which we develop a scalable approach to detecting novel documents in streams of text. We establish a sublinear regret bound, and empirically demonstrate orders of magnitude speedup over the batch algorithm, without much loss in performance. A further speedup can be achieved by distributing the algorithm using known techniques [3]. Apart from the target application of novel document detection, our online ℓ_1-dictionary learning algorithm could have broader applicability to other tasks using text and beyond, e.g., signal processing [8]. On a different note, there are several techniques that are related to dictionary learning, such as Latent Dirichlet Allocation [2], Probabilistic Latent Semantic Analysis [10], and Non-negative Matrix Factorization [12], and adapting these techniques for online detection of novel documents is a rich area for future work.

7. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE TSP*, 54(11), 2006.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 2011.

[4] V. Chenthamarakshan, P. Melville, V. Sindhwani, and R. D. Lawrence. Concept Labeling: Building Text Classifiers with Minimal Supervision. In *IJCAI*, 2011.

[5] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient Projections onto the l1-ball for Learning in High Dimensions. In *ICML*, pages 272–279, 2008.

[6] J. Duchi and Y. Singer. Efficient Online and Batch Learning using Forward Backward Splitting. *JMLR*, 10:2873–2898, 2009.

[7] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite Objective Mirror Descent. In *COLT*, pages 14–26, 2010.

[8] Q. Geng and J. Wright. On the Local Correctness of ℓ^1-Minimization for Dictionary Learning. Preprint: `http://www.columbia.edu/~jw2966`, 2011.

[9] E. Hazan, A. Agarwal, and S. Kale. Logarithmic Regret Algorithms for Online Convex Optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[10] T. Hofmann. Probabilistic Latent Semantic Analysis. In *UAI*, pages 289–296, 1999.

[11] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging Topic Detection using Dictionary Learning. In *CIKM*, pages 745–754, 2011.

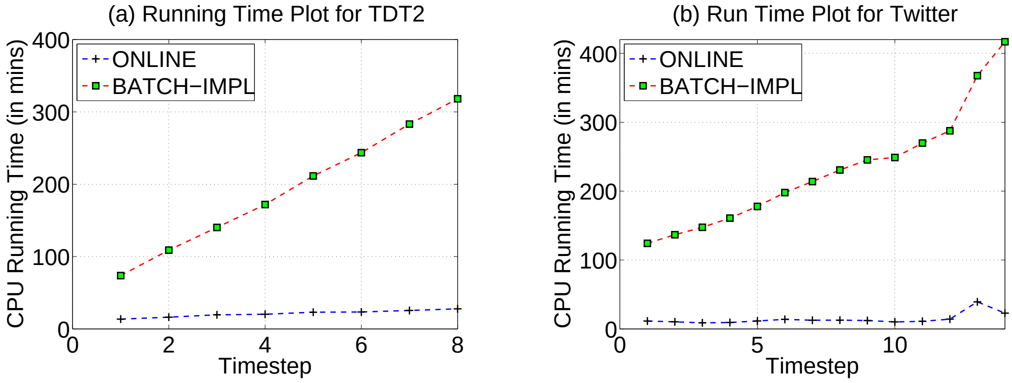

Figure 2: Running time plots for TDT2 and Twitter datasets.

Date	*Sample Novel Tweets Detected Using Algorithm* ONLINE
2011-09-26	Android powered 56 percent of smartphones sold in the last three months. Sad thing is it can't lower the rating of ios!
2011-09-29	How Windows 8 is faster, lighter and more efficient: WP7 Droid Bionic Android 2.3.4 HP TouchPad white ipods_72
2011-10-03	U.S. News: AT&T begins sending throttling warnings to top data hogs: AT&T did away with its unlimited da... #iPhone
2011-10-04	Can't wait for the iphone 4s #letstalkiphone
2011-10-05	Everybody put an iPhone up in the air one time #ripstevejobs

Table 2: Sample novel documents detected using Algorithm ONLINE **and some post-processing.**

[12] D. Lee and H. Seung. Learning the Parts of Objects by Non-negative Matrix Factorization. *Nature*, 1999.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *JMLR*, 2010.

[14] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] P. Melville, J. Leskovec, and F. Provost, editors. *Proceedings of the First Workshop on Social Media Analytics*. ACM, 2010.

[16] B. Olshausen and D. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[17] S. Petrović, M. Osborne, and V. Lavrenko. Streaming First Story Detection with Application to Twitter. In *ACL*, 2010.

[18] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer-Verlag, 2004.

[19] S. Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2), 2012.

[20] P. Tseng. Aprroximation Accuracy, Gradient Methods, and Error Bound for Structured Convex Optimization. *Mathematical Programming, Series B*, 125:263–295, 2010.

[21] H. Wang and A. Banerjee. Online Alternating Direction Method. In *ICML (to appear)*, 2012.

[22] J. Wright and Y. Ma. Dense Error Correction Via L1-Minimization. *IEEE TIT*, 56(7):3540–3560, 2010.

[23] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE TPAMI*, 2008.

[24] L. Xiao. Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. *JMLR*, 11:2543–2596, 2010.

[25] A. Y. Yang, S. S. Sastry, A. Ganesh, and Y. Ma. Fast L1-minimization Algorithms and an Application in Robust Face Recognition: A Review. In *ICIP*, pages 1849–1852, 2010.

[26] J. Yang and Y. Zhang. Alternating Direction Algorithms for L1-Problems in Compressive Sensing. *SIAM Journal of Scientific Computing*, 33(1):250–278, 2011.

[27] M. Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *ICML*, pages 928–936, 2003.