FAST ONLINE L_1-DICTIONARY LEARNING ALGORITHMS FOR NOVEL DOCUMENT DETECTION

Shiva Prasad Kasiviswanathan

General Electric Research, San Ramon, California, `kasivisw@gmail.com`

ABSTRACT

Online L_1-dictionary learning, introduced by Kasiviswanathan *et al.* [1], is the process of generating a sequence of (dictionary) matrices $\{A_{t+1}\}$, one at a time, for $t = 0, 1, \ldots$. After committing to A_{t+1}, a pair of matrices (P_{t+1}, X_{t+1}) is revealed and the online algorithm incurs a cost of $\|P_{t+1} - A_{t+1}X_{t+1}\|_1$. The goal of the online algorithm is to ensure that the total cost up to each time is not much larger than the smallest total cost of any fixed A chosen with the benefit of hindsight. In this paper, we study three different algorithms for this problem based on the schemes of dual averaging, projected gradient, and alternating direction method of multipliers. We focus on the performance of these algorithms for the application of novel document detection, where online dictionary learning could be used to automatically identify emerging topics of discussion from a voluminous stream of text documents in a scalable manner. Our empirical results show the relative benefits of these three algorithms for this application.

Index Terms- Dictionary Learning, Sparse Coding, Online Algorithms, Topic Detection

1. INTRODUCTION

In recent years, social media such as blogs and Twitter, are challenging traditional media outlets with their fast-paced dissemination of breaking news stories [2, 3, 4]. Given the high volume of online content generated, it is imperative to design near-real time technologies to distinguish documents belonging to *novel topics* from the background discussion. A document arriving at time t is considered *novel*, if the topic contained in the document is not already present in documents arrived until time $t-1$. Recently, Kasiviswanathan *et al.* [5] proposed a *dictionary learning* framework for detecting novel documents. Informally, the goal of dictionary learning is to find a dictionary matrix $A \in \mathbb{R}^{m \times k}$ such that each element $\mathbf{p}_i$ from a set of signals $[\mathbf{p}_1, \ldots, \mathbf{p}_n] \in \mathbb{R}^{m \times n}$ can be well-approximated as a (sparse) linear combination of the columns of A. Given a dictionary learning algorithm, we can use it to detect novel documents as follows: let A_t be a dictionary that can well-approximate all the documents arrived until time $t-1$, for a new data document $\mathbf{y}$ arriving at time t, if A_t can not well-approximate $\mathbf{y}$ then this indicates $\mathbf{y}$ is novel compared to documents in the past. We use an L_1-penalty on the loss function as it is a good choice for detecting novel documents [1]. This gives rise to an L_1-dictionary learning approach for novel document detection. The challenge then is to generate the sequence of dictionary matrices $A_1, A_2, \ldots$ in a scalable manner. A simple batch implementation is to generate the dictionary A_t using all the documents till time $t-1$ [5]. However, this batch approach is computationally infeasible as the optimization problems involved grow bigger with t. To overcome this issue, Kasiviswanathan *et al.* [1] proposed an online L_1-dictionary learning algorithm (based on the scheme of alternating directions method of multipliers) for generating these dictionaries and showed that it leads to a substantial speedup over the batch approach, without a loss of performance in detecting novel documents. We review their algorithm in Section 2.1.

Our Contributions. In this paper, we generalize the results of Kasiviswanathan *et al.* [1] by proposing a generic framework for novel document detection based on online L_1-dictionary learning. We instantiate this framework by using three algorithms for online L_1-dictionary learning based on the schemes of dual averaging [6], projected gradient [7], and alternating direction method of multipliers [8, 1]. Through extensive evaluation on two popular news-stream datasets, we compare the performance of these algorithms in detecting novel documents. Our experiments show that an online algorithm based on dual averaging has the best predictive performance and outperforms the alternating direction method of multipliers based algorithm presented in [1]. On the other hand, the algorithms based on projected gradient and alternating directions method of multipliers have some advantages over the dual averaging based algorithm in terms of running time and stability, respectively.

Notation. For a matrix M: $M^\top$ is its transpose, $\mathrm{sign}(M)$ is its sign matrix, $\|M\|_1 = \sum_{i,j} |m_{ij}|$, and $\|M\|_F^2 = \sum_{i,j} m_{ij}^2$. Let $\mathcal{A}$ be a convex set defined as: $\mathcal{A} = \{A \in \mathbb{R}^{m \times k} : A \geq 0,\ \forall j = 1, \ldots, k\ , \|A_j\|_1 \leq 1\}$, where A_j is the jth column in A. Let $\Pi_{\mathcal{A}}$ denote the projection onto the nearest point in $\mathcal{A}$, i.e., $\Pi_{\mathcal{A}}(M) = \mathrm{argmin}_{A \in \mathcal{A}}\ \|A - M\|_F^2$. This projection can be performed in near linear time [9].

Online Learning Model. In online learning an algorithm generates a sequence of decisions $x_{t+1} \in \mathcal{D}$ one at a time for $t = 0, 1, \ldots$ (where $\mathcal{D}$ is some domain). At time $t+1$, an unknown cost function g_{t+1} is revealed, and the algorithm encounters a loss of $g_{t+1}(x_{t+1})$. At the end of any time T, we define *regret* as the difference between the cumulative cost of the algorithm ($\sum_{t=0}^{T} g_{t+1}(x_{t+1})$) and the cost associated with best fixed decision $x \in \mathcal{D}$ from hindsight ($\min_{x \in \mathcal{D}} \sum_{t=0}^{T} g_{t+1}(x)$). The goal is to design algorithms whose 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.

2. L_1-DICTIONARY LEARNING

Dictionary learning concerns the problem of estimating a collection of basis vectors over which a given data collection can be accurately reconstructed, often with sparse encodings. The dictionary learning considers a finite set of signals $P = [\mathbf{p}_1, \dots, \mathbf{p}_n] \in \mathbb{R}^{m \times n}$ as input and optimizes the empirical cost function $f(A) = \sum_{i=1}^{n} \ell(\mathbf{p}_i, A)$, where $\ell(\cdot, \cdot)$ is some loss function and $A \in \mathbb{R}^{m \times k}$ is referred to as the dictionary.

In this paper, we follow the formulation from [5, 1, 10] and use an L_1-loss function with an L_1-regularization term. Kasiviswanathan *et al.* [1] have shown that in the context of novel document detection, imposing an L_1-loss results in a better scheme than imposing an L_2-loss as the L_1-loss better captures situations in text analysis where a term/phrase may become suddenly dominant in a discussion.[1] Therefore, for a signal (document) $\mathbf{p}_i$, $\ell(\mathbf{p}_i, A) = \min_{\mathbf{x}} \|\mathbf{p}_i - A\mathbf{x}\|_1 + \lambda \|\mathbf{x}\|_1$, where λ is the regularization parameter. This (sparse coding) formulation naturally takes into account both the error (with the $\|\mathbf{p}_i - A\mathbf{x}\|_1$ term) and the complexity of the sparse decomposition (with the $\|\mathbf{x}\|_1$ term). Using this in $f(A)$, we get the following dictionary learning formulation

$$\min_{A,X} \|P - AX\|_1 + \lambda \|X\|_1. \quad (1)$$

The matrix X is the coefficient matrix (also referred as the sparse code matrix). To prevent A from being arbitrarily large (which would lead to arbitrarily small values of X), we add a scaling constraint on A and require that each of its columns have L_1-norm less than equal to 1. In the novel document detection application, the matrices P, A, X will contain the associations between term-document, term-topic, and topic-document, respectively. More precisely, each column of A represents a topic and contains the contribution of the terms to that topic and similarly each row of X represents a topic and contains the contribution of the documents to that topic. Therefore, for maintaining the interpretability of term-topic and topic-document associations, we require A and X to be non-negative (this is also known as non-negative matrix factorization, [13]). These additions to (1) gives:

$$\min_{A \in \mathcal{A}} f(A) = \min_{A \in \mathcal{A}, X \geq 0} \|P - AX\|_1 + \lambda \|X\|_1. \quad (2)$$

This problem is jointly non-convex in (A, X) and the standard offline approach for solving it is to alternatively update X and A until some convergence criteria is met.

2.1. Online L_1-Dictionary Learning

In this section, we define the online L_1-dictionary learning problem and provide algorithms for solving it.[2] Since the optimization problem in (2) is non-convex, it may not be possible to design polynomial time offline algorithms to solve it without making any assumptions (on either A or X). This also means that it may not be possible to design a polynomial time online algorithm with sublinear regret for (2) without making any assumptions because that would imply a polynomial time offline algorithm for solving (2). Therefore, we work with the following relaxation of the problem, where the focus is to obtain regret bounds for updating the dictionary, assuming that the sparse code matrices are also part of the input.

Definition 1 (Online L_1-Dictionary Learning (ODL) Problem [1]). [3] *At time $t = 0, 1, \dots,$ the online algorithm picks $A_{t+1} \in \mathcal{A}$. Then, the nature reveals (P_{t+1}, X_{t+1}) with $P_{t+1} \in \mathbb{R}^{m \times n}$ and $X_{t+1} \in \mathbb{R}^{k \times n}$. The problem is to pick the $\{A_t\}$ sequence such that the following regret function is minimized*

$$\sum_{t=0}^{T} \|P_{t+1} - A_{t+1}X_{t+1}\|_1 - \min_{A \in \mathcal{A}} \sum_{t=0}^{T} \|P_{t+1} - AX_{t+1}\|_1.$$

The convexity of the cost function in the ODL problem enables development of optimization-based algorithms for obtaining sublinear regret. In this paper, we design three different algorithms for the ODL problem based on: (i) *dual averaging scheme* of Nesterov [6], (ii) *projected gradient scheme* of Zinkevich [7], and (iii) *online alternating directions scheme* of Wang and Banerjee [8]. Theoretically, all three algorithms achieve a regret of $O(\sqrt{T})$ for the ODL problem. However, as we show in Section 3.1, their practical performances for the application of novel document detection can vary quite widely. This happens because novel document detection requires solving a non-convex problem (Equation (2)), and we do not have any guarantees on reaching a global optima using (any of) the ODL algorithms. A detailed discussion is deferred to the full version of this paper.

(1) Dual Averaging Scheme. Our first ODL algorithm (referred as the DA algorithm) is based on adapting Nesterov's dual averaging scheme [6, 15]. [4] The dual averaging scheme is based on a *proximal function* $\psi : \mathcal{A} \to \mathbb{R}$ assumed to be 1-strongly convex with respect to some norm

[1] Similarly, use of L_1-loss have also led to more robust face recognition algorithms [11, 12].

[2] Rao and Porikli [14] define online dictionary learning in a different context, where the goal is to dynamically adjust the dictionary size (k) to the incoming data. Their algorithm does not have a regret guarantee and works for the L_2-loss.

[3] The setting considered in [1] is slightly more general. The techniques developed in this paper also apply to that general setting.

[4] The dual averaging scheme is similar to the *follow the perturbed leader* approaches developed in the online optimization [16] community, though the specific approach that we use here is due to Nesterov [6].

$\|\cdot\|$. We use the standard quadratic proximal function $\psi(A) = (1/2)\|A\|_F^2$. The DA algorithm makes a prediction A_{t+1} using the average subgradient till time t. More concretely, the DA algorithm generates a sequence of iterates $\{A_t, \widehat{G}_t\}$, where $A_t \in \mathcal{A}$ and $\widehat{G}_t$ is the sum of subgradients as defined below, using the following steps. At time t, the algorithm computes the subgradient of $\|P_t - AX_t\|_1$ evaluated at $A = A_t$. Let G_t denote this subgradient, G_t equals

$$\frac{\partial}{\partial A}\|P_t - AX_t\|_1\bigg|_{A=A_t} = \operatorname{sign}(A_tX_t - P_t)X_t^\top,$$

and then performs the updates $\widehat{G}_{t+1} = \widehat{G}_t + G_t$ and

$$A_{t+1} = \operatorname{argmin}_{A\in\mathcal{A}} \frac{1}{t}\langle \widehat{G}_{t+1}, A\rangle + \frac{\gamma}{2\sqrt{t}}\|A\|_F^2, \qquad (3)$$

where $\gamma > 0$ is a parameter to the algorithm. The A_{t+1} has a closed-form update given by:

$$A_{t+1} = \Pi_{\mathcal{A}}(-\widehat{G}_{t+1}/(\gamma\sqrt{t})).$$

The underlying intuition here is to pick A_{t+1} to minimize an averaged first-order approximation to the cost function, while the second term in (3) enforce that the iterates not oscillate wildly.

(2) Projected Gradient Scheme. Our next ODL algorithm (referred as the PG algorithm) is based on the classical projected gradient scheme of Zinkevich [7]. The general idea of projected gradient scheme is to generate a sequence of iterates by taking a descent step in the negative gradient direction and then project the result onto the constraint set. More concretely, the PG algorithm at time t performs the following update[5]

$$A_{t+1} = \Pi_{\mathcal{A}}\,(A_t - \eta\cdot G_t),$$

where $\eta > 0$ is the step size of the algorithm.

(3) Online Alternating Directions Scheme. Our last ODL algorithm (referred as the ADMM algorithm) is based on the scheme of alternating direction method of multipliers, and was recently proposed by Kasiviswanathan *et al.* [1]. We give here a brief review of the algorithm from [1]. The general idea of alternating directions method of multipliers is to minimize the augmented Lagrangian function using a Gauss-Seidel type update of both the primal and dual variables. Consider the following minimization problem $\min_{A\in\mathcal{A}}\|P_t - AX_t\|_1$. We can rewrite this above minimization problem as:

$$\min_{A\in\mathcal{A},\Gamma}\|\Gamma\|_1 \text{ such that } \Gamma = P_t - AX_t. \qquad (4)$$

The augmented Lagrangian of this problem is:

$$\|\Gamma\|_1 + \langle\Delta_t, P_t - AX_t - \Gamma\rangle + \frac{\alpha}{2}\|P_t - AX_t - \Gamma\|_F^2,$$

where $A \in \mathcal{A}, \Gamma \in \mathbb{R}^{m\times n}$ are the primal variables and $\Delta \in \mathbb{R}^{m\times n}$ is the dual variable. At time t, the algorithm updates $\{\Gamma_t, A_t, \Delta_t\}$ as follows:

$$\widetilde{\Gamma}_t = P_t - A_tX_t,$$

$$\Gamma_{t+1} = \operatorname{sign}(\widetilde{\Gamma}_t + \Delta_t/\alpha)\cdot\max\{|\widetilde{\Gamma}_t + \Delta_t/\alpha| - 1/\alpha, 0\},$$

$$A_{t+1} = \Pi_{\mathcal{A}}\left(\max\left\{0, A_t + \frac{(\Delta_t/\alpha + \widetilde{\Gamma}_t - \Gamma_{t+1})X_t^\top}{(2\Psi_{\max}(X_t))}\right\}\right),$$

$$\Delta_{t+1} = \Delta_t + \alpha(P_t - A_{t+1}X_t - \Gamma_{t+1}),$$

where $\Psi_{\max}(X_t)$ is the maximum eigenvalue of $X_t^\top X_t$ and $\alpha > 0$ is a parameter to the algorithm. At any timestep t, the ADMM algorithm could violate the equality constraint in (4) (i.e., $\Gamma_{t+1} \neq P_t - A_{t+1}X_t$), but as shown by Kasiviswanathan *et al.* [1] the equality constraint is satisfied on average in the long run (more formally, an $O(\sqrt{T})$ regret can be established on the equality constraint violation, see [1] for more details).

3. NOVEL DOCUMENT DETECTION

Let $\{P_t : P_t \in \mathbb{R}^{m\times n}, t = 1, 2, 3, \dots\}$ denote a sequence of matrices where each column of P_t represents a document arriving at time t.[6] Each document is represented in the TF-IDF vector space model [18], and we normalize P_t such that each column (document) in P_t has a unit L_1-norm. The *goal* of novel document detection is to identify documents (in P_t) that contains topics not present in the documents in $[P_1|\dots|P_{t-1}]$.

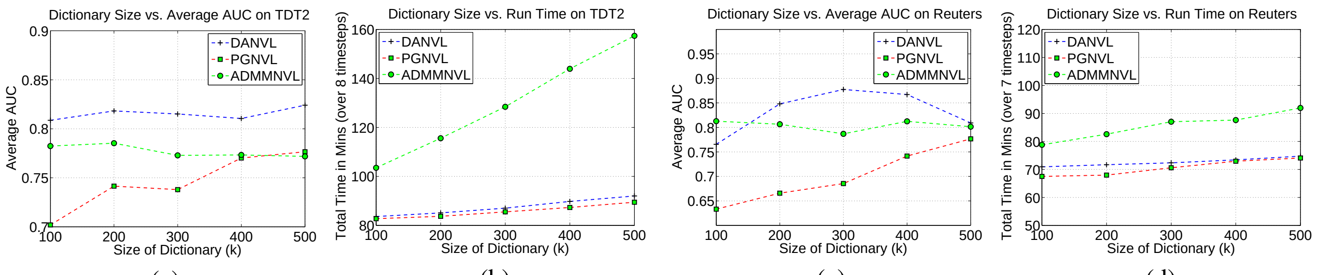
Online Algorithm to Detect Novel Documents. We now describe a generic mechanism that uses an ODL algorithm $\mathbb{A}$ to detect novel documents. At each time step t, Mechanism GENNVL alternates between a novel document detection and an online dictionary learning stage. The novel document detection stage solves the sparse coding problem (with the added constraint $\mathbf{x} \geq 0$) for each document in P_t. A threshold ζ is used to mark a document as novel or non-novel. The performance of this procedure depends on the ability of the dictionary A_t to *accurately* reconstruct all the documents in $[P_1|\dots|P_{t-1}]$, or in other words to accurately represent all the term-topic associations contained in the documents in $[P_1|\dots|P_{t-1}]$. The dictionaries are generated using $\mathbb{A}$. Based on the choice of $\mathbb{A}$, we get different instantiations of this generic mechanism. We refer to instantiations of Mechanism GENNVL with the DA, PG, and ADMM algorithms as DANVL, PGNVL, and ADMMNVL, respectively. Note that the sequence of $\{X_t\}$ and $\{A_t\}$ matrices generated by these algorithms could be quite different.

3.1. Experimental Results

In our experiments, we compare the performance of DANVL, PGNVL, and ADMMNVL algorithms on two human-labeled

[5] A similar update step could also be derived by starting from the forward-backward splitting algorithm of Duchi and Singer [17].

[6] For simplicity in exposition, we will assume m and n are independent of t. This is without loss of generality as we can always zero-pad the matrices appropriately.



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

Fig. 1: AUC and execution time plots for the TDT2 and Reuters datasets.

Mechanism GenNvl: Generic Novel Document Detection using an Odl Algorithm $\mathbb{A}$

```
1: Input: $P_t = [\mathbf{p}_1, \dots, \mathbf{p}_n]$, $A_t$, $\lambda$, $\zeta$
2: Novel Document Detection Stage:
3: For $i = 1$ to $n$ do
4:     If $(\min_{\mathbf{x} \geq 0} \|\mathbf{p}_i - A_t\mathbf{x}\|_1 + \lambda\|\mathbf{x}\|_1 \geq \zeta)$
5:         Mark $\mathbf{p}_i$ as novel
6: Dictionary Learning Stage:
7: $X_t = \min_{X \geq 0} \|P_t - A_t X\|_1 + \lambda\|X\|_1$
8: Generate $A_{t+1}$ by using the Odl algorithm $\mathbb{A}$
```

text datasets that are commonly used in topic modeling literature: TDT2 and Reuters. The TDT2 dataset consists of a set of 9000 documents represented over 19528 terms, and the Reuters dataset consists of a set of 8000 documents represented over 18933 terms.

Evaluation Metric. For performance evaluation, we assume that documents in the corpus have been manually identified with a set of topics. The *true* label of a document $\mathbf{y}$ arriving at time t is *novel* if the (dominant) topic of $\mathbf{y}$ has not appeared before the time t. The task of novel document detection is to *classify* each document as either novel or non-novel. For evaluating this classification task, we use the Area Under the ROC Curve (AUC) [18].

Experimental Setup. All reported results are based on a Matlab implementation running on a 2.5 GHz Intel processor with 8GB RAM. Sparse coding (Step 4 of Mechanism GenNvl) problem is solved using an alternating directions method of multipliers algorithm (refer [5] for more details). The regularization parameter λ is set to 0.1 which yields reasonable sparsities in our experiments. We introduce the documents in groups. We initialize the dictionary using the first 1000 documents by alternatively repeating lines 7 and 8 of Mechanism GenNvl 30 times (no evaluation is done here). In each subsequent timestep, we evaluate the mechanism by providing it with the next set of 1000 documents, followed by dictionary update.

Setting the Parameters and Sensitivity to them. The parameters of DaNvl, PgNvl, and AdmmNvl are γ, η, and α, respectively. In Figure 2, we examine performance of these algorithms under various setting of their parameters. On Y-axis we plot the average AUC (obtained by averaging AUC's over all timesteps). Notice that the performances of both DaNvl and PgNvl are quite sensitive to changes in their respective parameter values, whereas performance of AdmmNvl is much more robust to its parameter value. We set $\gamma = 4$, $\eta = 11$, and $\alpha = 9$ in our experiments based on Figure 2.

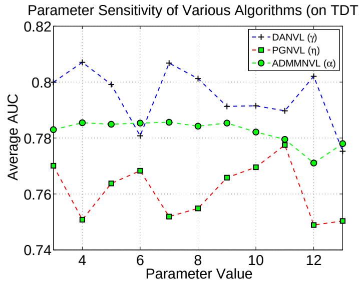


Fig. 2: Parameter sensitivity on TDT2 ($k = 200$).

Results on TDT2 and Reuters Datasets. Figure 1 shows the performance of the algorithms on the TDT2 and Reuters datasets. We treat the dictionary size (k) as a variable in these plots and vary it from 100 to 500 in increments of 100. On the AUC metric, DaNvl outperforms the other two algorithms (see, Figures 1(a) and 1(c)), e.g., on the TDT2 dataset at $k = 500$, DaNvl is 6.7% better than either PgNvl or AdmmNvl. The performance of AdmmNvl has a smaller variation with k than the other two algorithms and it performs well when dictionary size is small ($k = 100$), whereas on the other hand the performance of PgNvl improves with k.[7]

Running times of all the three algorithms increase with k because of the increase in the cost of matrix multiplications involved. Among the three algorithms, AdmmNvl is the slowest, and its running time also increases at a faster rate as a function of k. Both of the gradient-based algorithms DaNvl and PgNvl have similar run time profiles, with PgNvl being slightly faster than DaNvl.

Conclusion. DaNvl has the best predictive performance, whereas PgNvl and AdmmNvl have some advantages in terms of running time and parameter stability, respectively.

[7] Since Mechanism GenNvl performs only online update of the dictionaries, a larger k may not lead to a better AUC performance.

4. REFERENCES

[1] S. P. Kasiviswanathan, H. Wang, A. Banerjee, and P. Melville, "Online ℓ_1-Dictionary Learning with Application to Novel Document Detection," in *NIPS*, 2012.

[2] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo, "Twitter Under Crisis: Can we Trust What we RT?," in *Workshop on Social Media Analytics*, 2010.

[3] Zicong Zhou, Roja Bandari, Joseph Kong, Hai Qian, and Vwani Roychowdhury, "Information resonance on twitter: watching iran," in *Workshop on Social Media Analytics*, 2010.

[4] Michael Mathioudakis and Nick Koudas, "Twittermonitor: trend detection over the twitter stream," in *SIGMOD*. 2010, pp. 1155–1158, ACM.

[5] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani, "Emerging Topic Detection using Dictionary Learning," in *CIKM*, 2011, pp. 745–754.

[6] Yurii Nesterov, "Primal-dual Subgradient Methods for Convex Problems," *Math. Program.*, vol. 120, no. 1, pp. 221–259, 2009.

[7] Martin Zinkevich, "Online Convex Programming and Generalized Infinitesimal Gradient Ascent," in *ICML*, 2003, pp. 928–936.

[8] H. Wang and A. Banerjee, "Online Alternating Direction Method," in *ICML*, 2012.

[9] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra, "Efficient Projections onto the l1-ball for Learning in High Dimensions," in *ICML*, 2008, pp. 272–279.

[10] S.P. Kasiviswanathan, G. Cong, P. Melville, and R. Lawrence, "Novel Document Detection on Massive Data Streams using Distributed Dictionary Learning," *To appear in IBM Journal of Research and Development*, 2013.

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

[12] J. Wright and Y. Ma, "Dense Error Correction Via L1-Minimization," *IEEE TIT*, vol. 56, no. 7, pp. 3540–3560, 2010.

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

[14] N. Rao and F. Porikli, "A Clustering Approach to Optimize Online Dictionary Learning," in *ICASSP*, 2012, pp. 1293–1296.

[15] Lin Xiao, "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization," *JMLR*, pp. 2543–2596, 2010.

[16] Adam Kalai and Santosh Vempala, "Efficient Algorithms for Online Decision Problems," *JCSS*, vol. 71, no. 3, pp. 291–307, 2005.

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

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