Online Dictionary Learning from Large-Scale Binary Data

Yanning Shen and Georgios B. Giannakis

Dept. of ECE and DTC, University of Minnesota, Minneapolis, USA
Emails: {shenx513, georgios}@umn.edu

***Abstract*—Compressive sensing (CS) has been shown useful for reducing dimensionality, by exploiting signal sparsity inherent to specific domain representations of data. Traditional CS approaches represent the signal as a sparse linear combination of basis vectors from a prescribed dictionary. However, it is often impractical to presume accurate knowledge of the basis, which motivates data-driven dictionary learning. Moreover, in large-scale settings one may only afford to acquire quantized measurements, which may arrive sequentially in a streaming fashion. The present paper jointly learns the sparse signal representation and the unknown dictionary when only binary streaming measurements with possible misses are available. To this end, a novel efficient online estimator with closed-form sequential updates is put forth to recover the sparse representation, while refining the dictionary 'on the fly'. Numerical tests on simulated and real data corroborate the efficacy of the novel approach.**

***Index Terms*—dictionary learning, binary data, online learning.**

I. Introduction

Most recovery approaches to compressive sensing (CS) typically assume that sampled measurements have infinite precision, but observed measurements are always quantized in practice. Moreover, due to the simple and efficient hardware implementation of one bit quantizers, there has been a rising interest in recovering sparse representations based on binary measurements; see e.g., [2], [6]. Existing works on one-bit CS, assume that the dictionary is known a priori. Although certain off-the-shelf dictionaries such as the Fourier or wavelet bases yield good performance in several applications, it has been shown that, a data-driven approach which learns the dictionary from the data can improve the recovery performance; see e.g., [1], [3] and references therein. Extensions of the dictionary learning (DL) paradigm to one-bit measurements have recently been studied in [5] and [16]. The batch complexity penalized maximum likelihood estimator was developed in [4]; see also [16] for a batch iterative algorithm.

In large-scale settings, where new data are often acquired sequentially in a streaming fashion. For example, in recommender systems, there are millions of 'like's and 'dislike's for sequentially released movies, and newly released ones call for real-time recommendation. Furthermore, acquired data may contain misses, since e.g., user ratings for a big portion of movies are missing. In general however, most one-bit CS and DL approaches are available for batch operator, assuming that all data are available beforehand. For instance, the advocated approach in [16] is not tailored to handle misses or streaming data. Towards developing online DL algorithms, the stochastic approximation framework has been shown useful when measurements are assumed to have infinite precision [9], [14]. The present paper broadens the merits of these prior works to binary DL from streaming data with misses. Specifically, a batch estimator with closed-form updates will first be developed, and then its online version that leverages stochastic gradient descent iterations will be put forth. Both algorithms are provably convergent, and can explicitly handle and impute misses.

To place this work in context, binary principle component analysis (PCA), which seeks a lower-dimensional sketch of one-bit data has been developed in e.g., [8], while its online renditions have been advocated in [7] and [13]. It is worth noting that binary PCA requires the lower dimensional representations to lie in the same lower dimensional subspace. However, binary DL is more general in the sense that it does not enforce this constraint. In fact, the sparse representations are allowed to have varying sparsity patterns. Naturally, binary DL has the potential to reveal sparse representers of different subspaces, thus subsuming inference tasks such as subspace clustering. DL using one-bit data is also useful for binary classification problem with intentionally or unintentionally missing features.

II. Preliminaries and Problem Statement

Consider the high-dimensional $M \times 1$ vectors $\{\mathbf{y}_\tau \in \{-1,1\}^M\}_{\tau=1}^T$ corresponding to sign measurements

$$\mathbf{y}_\tau = \text{sign}(\mathbf{D}\mathbf{s}_\tau + \mathbf{n}_\tau) \tag{1}$$

where $\{\mathbf{s}_\tau \in \mathbb{R}^N\}_{\tau=1}^T$ are sparse vectors with $K \ll N$ non-zeros entries, and $\mathbf{n}_\tau$ captures unmodeled dynamics, while $\mathbf{D} \in \mathbb{R}^{M\times N}$ denotes the unknown dictionary. Supposing that some measurements are missing, let $\Omega_\tau \subseteq \{1,\ldots,M\}$ denote the index set of available measurements at time τ, with $|\Omega_\tau| \le M$. For instance, in recommender systems $\{\mathbf{y}_{i\tau}\}_{i\in\Omega_\tau}$ represent the available binary ratings, namely, "like" or "dislike"). In binary classification problems, $\{y_{i\tau},\ i \in \Omega_\tau\}_{\tau=1}^T$ corresponds to the available labels.

Given possibly partial observations $\{y_{i\tau},\ i \in \Omega_\tau\}_{\tau=1}^T$, the goal of the present paper is to recover $\{\mathbf{s}_\tau\}_{\tau=1}^T$ and $\mathbf{D}$. So that the missing entries can be imputed and ensuing

Work in this paper was supported by NSF 1514056.

inference tasks such as subspace clustering can be carried out based on $\{\mathbf{s}_\tau\}$. Based on (1), the i-th entry of $\mathbf{y}_\tau$ is

$$y_{i\tau} = \text{sign}(\mathbf{d}_i^\top \mathbf{s}_\tau + n_{i\tau}), \quad i \in \Omega_\tau \tag{2}$$

where $\mathbf{d}_i^\top$ denotes the ith row of $\mathbf{D}$. If $n_{it} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$, the entry-wise likelihood function can be written as

$$\begin{aligned} &\ell(y_{i\tau}; \mathbf{d}_i, \mathbf{s}_\tau) \\ &= \left[\Pr(n_{i\tau} > -\mathbf{d}_i^\top \mathbf{s}_\tau)\right]^{\frac{y_{it}+1}{2}} \Pr(n_{i\tau} < -\mathbf{d}_i^\top \mathbf{s}_\tau)^{\frac{1-y_{it}}{2}} \\ &= \left[Q\left(-\mathbf{d}_i^\top \mathbf{s}_\tau/\sigma_n\right)\right]^{\frac{y_{it}+1}{2}} \left[Q\left(\mathbf{d}_i^\top \mathbf{s}_\tau/\sigma_n\right)\right]^{\frac{1-y_{it}}{2}} \\ &= Q\left(-y_{it}\mathbf{d}_i^\top \mathbf{s}_\tau/\sigma_n\right) \end{aligned} \tag{3}$$

where $Q(\cdot)$ denotes the standard Gaussian tail function, see e.g., [11], [13]. With $\mathbf{S} := [\mathbf{s}_1, \ldots, \mathbf{s}_T]$, the log-likelihood of the available binary data can be written as

$$\begin{aligned} \log \mathcal{L}\Big(\{y_{i\tau},\ i \in \Omega_\tau\}_{\tau=1}^T; \mathbf{D}, \mathbf{S}\Big) \\ = \sum_{\tau=1}^T \sum_{i \in \Omega_\tau} \log \ell(y_{i\tau}; \mathbf{d}_i, \mathbf{s}_\tau). \end{aligned} \tag{4}$$

Since $\mathbf{s}_\tau$ is sparse, a natural regularizer to promote sparsity is the ℓ_0-norm, which counts the number of non-zero entries. However, minimizing an ℓ_0-norm penalized cost incurs NP complexity, and contemporary approaches resort to the ℓ_1-norm as the closest convex relaxation. To this end, one is motivated to minimize the regularized log-likelihood

$$\min_{\mathbf{D}\in\mathcal{D}, \mathbf{S}\in\mathcal{S}} - \log \mathcal{L}\Big(\{y_{i,\tau},\ i \in \Omega_\tau\}_{\tau=1}^T; \mathbf{D}, \mathbf{S}\Big) + \lambda \sum_{\tau=1}^T \|\mathbf{s}_\tau\|_1 \tag{5}$$

where $\|\mathbf{s}_\tau\|_1 := \sum_{i=1}^N |s_{i\tau}|$, and the constraint set $\mathcal{D} := \{\mathbf{D} \in \mathbb{R}^{M\times N} :\ \|\mathbf{d}_i\|_2 \leq 1,\ \forall i = 1, \cdots, M\}$ prevents the entries of $\mathbf{D}$ from taking on arbitrarily large values. If $\mathbf{D}$ were unconstrained, one would run the risk of trivial solutions, with $\mathbf{s}_\tau$ approaching $\mathbf{0}$ due to the ℓ_1-norm regularizer. Furthermore, since the sign operator in (2) suppresses magnitude information, there is no loss of generality by a fortiori setting $\|\mathbf{s}_\tau\|_2 = 1,\ \forall\tau$. This justifies constraining the matrix $\mathbf{S}$ to the set $\mathcal{S} := \{\mathbf{S} \in \mathbb{R}^{N\times T} :\ \|\mathbf{s}_\tau\|_2 \leq 1,\ \forall \tau = 1, \ldots, T\}$.

For moderate values of M and T, one can devise a batch alternating minimization (AM) scheme if the entire dataset is available. This is tantamount to alternately minimizing (5) with respect to (w.r.t.) $\mathbf{D}$ while holding $\mathbf{S}$ fixed, followed by minimization of $\mathbf{S}$ with $\mathbf{D}$ fixed, in an iterative manner until convergence is attained. Note that (5) decouples over both i and τ, and the resulting subproblems involve minimization of $\ell(y_{i,t}; \mathbf{d}_i, \mathbf{s}_t)$ w.r.t. i and τ. Each subproblem does not lead to a closed-form solution, and one must resort to iterative approaches (e.g., gradient descent). This is challenging in big data settings, when M and T are large, and $\mathbf{y}_\tau$ may be acquired sequentially in a streaming fashion ($T \to \infty$). In this case, a real-time algorithm for solving (5) is well motivated, and iterative solvers that must attain convergence per acquired datum are impractical. In order to process streaming data, the sequel will first introduce an efficient, provably convergent batch algorithm, based on a modified objective function. An online rendition of the algorithm, that adopts first-order stochastic gradient descent iterations will then be developed to process large-scale streaming data.

III. BINARY DICTIONARY LEARNING

A. One-bit batch algorithm

Further inspection of $\ell(y_{it}, \mathbf{d}_i, \mathbf{s}_t)$ in (3) shows that it is a monotonically increasing function of $y_{it}\mathbf{d}_i^\top \mathbf{s}_t$. Consequently, $y_{it}\mathbf{d}_i^\top \mathbf{s}_t$ can be adopted as a metric that capturing how well y_{it} is represented. Based on this observation, one is motivated to solve the following optimization problem

$$\{\hat{\mathbf{D}}, \hat{\mathbf{S}}\} := \arg \min_{\mathbf{D}\in\mathcal{D}, \mathbf{S}\in\mathcal{S}} \sum_{\tau=1}^T \left[- \sum_{i\in\Omega_\tau} y_{i\tau}\mathbf{d}_i^\top \mathbf{s}_\tau + \lambda \|\mathbf{s}_\tau\|_1\right] \tag{6}$$

instead of (5). Even though (6) is not equivalent to (5), it can be shown that if $\mathbf{D}$ is available, then solving (6) w.r.t. $\mathbf{S}$ can yield reliable reconstruction performance under reasonable conditions [17].

Note that (6) is block multi-convex [12] w.r.t. to $\mathbf{D}$ and $\mathbf{S}$, that is, it is convex w.r.t. one block of variables when the others are fixed. Block multi-convex problems can be solved using a block coordinate descent (BCD) iteration, with convergence guarantees to a stationary point [15]. The present paper advocates BCD iterations, which amount to the following updates per iteration k.

$$\mathbf{S}[k] := \arg \min_{\mathbf{S}\in\mathcal{S}} \sum_{\tau=1}^T \left[\sum_{i\in\Omega_\tau} -y_{i\tau}\mathbf{d}_i^\top[k-1]\mathbf{s}_\tau + \lambda\|\mathbf{s}_\tau\|_1\right] \tag{7a}$$

$$\mathbf{D}[k] := \arg \min_{\mathbf{D}\in\mathcal{D}} \sum_{t=1}^T \sum_{i\in\Omega_\tau} -y_{i\tau}\mathbf{d}_i^\top \mathbf{s}_\tau[k]\ . \tag{7b}$$

Note that (7a) is separable across columns of $\mathbf{S}$, with the corresponding subproblems admitting closed-form solutions

$$\mathbf{s}_\tau[k] = \begin{cases} \mathbf{0}, & \|\mathbf{D}[k-1]\tilde{\mathbf{y}}_\tau\|_\infty \leq \lambda \\ \frac{P_\lambda(\mathbf{D}^\top[k-1]\tilde{\mathbf{y}}_\tau)}{\|P_\lambda(\mathbf{D}[k-1]^\top \tilde{\mathbf{y}}_\tau)\|_2}, & \text{otherwise} \end{cases} \tag{8}$$

where $\tilde{\mathbf{y}}_\tau := [\tilde{y}_{1\tau}, \ldots, \tilde{y}_{M\tau}]^\top \in \mathbb{R}^M$ with $\tilde{y}_{i\tau} = y_{i\tau}, \forall i \in \Omega_\tau$, otherwise $\tilde{y}_{i\tau} = 0$; while the entry-wise operator $P_\lambda(\cdot)$ is

$$P_\lambda(z) := \begin{cases} 0, & \text{if } z \leq \lambda \\ \text{sign}(z)(|z| - \lambda), & \text{otherwise.} \end{cases} \tag{9}$$

Similarly, (7b) is separable across rows of $\mathbf{D}$; that is,

$$\mathbf{d}_i[k] = \arg \min_{\|\mathbf{d}_j\|\leq 1} \sum_{\tau=1}^T -\tilde{y}_{i\tau}\mathbf{d}_i^\top \mathbf{s}_\tau[k] \tag{10}$$

Algorithm 1 One-bit batch dictionary learning algorithm

input: $\{\mathbf{y}_t\}_{\tau=1}^T$, λ
initialize: $\mathbf{D}[0]$
for $k = 1, 2, \ldots$ **do**
  (S1) For $\tau = 1, \cdots, T$, update $\mathbf{s}_\tau[k]$ via (8).
  (S2) Dictionary refinement via (11).
end for
return $\hat{\mathbf{D}} = \mathbf{D}[k]$

Algorithm 2 One-bit online dictionary learning algorithm

input: $\{\mathbf{y}_t\}_{\tau=1}^T$, λ
initialize: $\hat{\mathbf{D}}_0$
for $t = 1, 2, \ldots T$ **do**
  (S1) Sparse coding: Update $\mathbf{s}_t$ via (14).
  (S2) Dictionary refinement: Update $\hat{\mathbf{D}}_t$ via(16).
end for
return $\hat{\mathbf{D}} = \hat{\mathbf{D}}_t$

which leads to the following closed-form solution

$$\mathbf{d}_i[k] = \frac{\sum_{\tau=1}^T -\tilde{y}_{i\tau}\mathbf{s}_\tau[k]}{\| \sum_{\tau=1}^T -\tilde{y}_{i\tau}\mathbf{s}_\tau[k]\|_2}. \tag{11}$$

Algorithm 1 summarizes the developed batch iterative scheme for dictionary learning from binary data.

B. One-bit online algorithm

In order to attain real-time operation for large-scale streaming settings, this subsection deals with data that are acquired sequentially. To this end, recast (6) to minimize the following expected cost per t

$$\min_{\mathbf{s}_t, \mathbf{D}} \quad \mathbb{E}\left\{g_t\big(\{y_{it}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D}\big)\right\} \tag{12}$$

where

$$g_t\big(\{y_{i,t}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D}\big) := -\sum_{i\in\Omega_t} y_{it}\mathbf{d}_i^\top \mathbf{s}_t + \lambda\|\mathbf{s}_t\|_1$$

and expectation in (12) is taken w.r.t. the unknown probability distribution of $\{y_{it}\}$. To solve (12), one can approximate the expectation as $\mathbb{E}\left\{g_t\big(\{y_{it}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D}\big)\right\} \approx (1/t)\sum_{\tau=1}^t \left\{g_\tau\big(\{y_{i\tau}\}_{i\in\Omega_\tau}; \mathbf{s}_\tau, \mathbf{D}\big)\right\}$, which accumulates all past data up until t. Instead the simple instantaneous approximation

$$\mathbb{E}\left\{g_t\big(\{y_{it}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D}\big)\right\} \approx g_t\big(\{y_{it}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D}\big) \tag{13}$$

which discards all past data and leads to computationally affordable updates.

Minimizing (13) per t can be accomplished via AM iterations along the lines of [13], with the iteration index coinciding with t. This scheme comprises two steps per t, upon acquisition of $\{y_{i,t}\}_{i\in\Omega_t}$. First, the sparse vector $\mathbf{s}_t$ is recovered from the incomplete binary measurements, by solving

$$\textbf{(S1)} \quad \hat{\mathbf{s}}_t = \arg\min_{\mathbf{s}\in\mathbb{R}^d} g_t\big(\{y_{i,t}\}_{i\in\Omega_t}; \mathbf{s}, \hat{\mathbf{D}}_{t-1}\big) \tag{14}$$

with $\mathbf{D}$ set to the most recent update $\hat{\mathbf{D}}_{t-1}$. Step (**S1**) entails convex minimization which admits the closed-form solution

$$\hat{\mathbf{s}}_t = \begin{cases} \mathbf{0}, & \|\hat{\mathbf{D}}_{t-1}\tilde{\mathbf{y}}_t\|_\infty \leq \lambda \\ \frac{P_\lambda\left(\hat{\mathbf{D}}_{t-1}^\top \tilde{\mathbf{y}}_t\right)}{\|P_\lambda\left(\hat{\mathbf{D}}_{t-1}^\top \tilde{\mathbf{y}}_t\right)\|_2}, & \text{otherwise.} \end{cases} \tag{15}$$

In the subsequent step, the dictionary is refined based on $\hat{\mathbf{s}}_t$, by a single projected stochastic gradient descent (SGD) iteration given by

$$\textbf{(S2)} \quad \hat{\mathbf{D}}_t = \text{proj}_{\mathcal{D}}\left(\hat{\mathbf{D}}_{t-1} - \mu_t \nabla_{\mathbf{D}} g_t\big(\{y_{i,t}\}_{i\in\Omega_t}; \hat{\mathbf{s}}_t, \mathbf{D}\big)\right) \tag{16}$$

where μ_t is the step size. The projection operator $\text{proj}_{\mathcal{D}}(\cdot)$ is available row-wise in the closed-form

$$\text{proj}_{\mathcal{D}}(\mathbf{d}_\mathrm{i}) = \mathbf{d}_\mathrm{i}/\max\{\|\mathbf{d}_\mathrm{i}\|_2,\ 1\} \tag{17}$$

while the gradient can be readily found as

$$\nabla_{\mathbf{D}} f_t(\hat{\mathbf{s}}_t, \mathbf{D}) = -\tilde{\mathbf{y}}_t \hat{\mathbf{s}}_t^\top. \tag{18}$$

Algorithm 2 summarizes the developed online scheme based on SGD iterations.

Remark 1 [Computational cost]. The dictionary update in Algorithm 2 is parallelizable, while the sparse signal recovery admits closed-form updates. Indeed, the developed online algorithm entails computationally affordable iterations, which scale to large datasets.

Remark 2 [Convergence]. As mentioned earlier, the objective function is block multi-convex, hence the convergence analysis in [9], [10] can be adopted to establish asymptotic convergence of Algorithm 2 under certain conditions (skipped due to space limitations). Nevertheless, extensive numerical tests presented in Section IV, empirically confirm convergence of the online algorithm.

IV. Numerical Tests

This section assesses the recovery performance of the developed algorithms on both simulated and real data.

A. Synthetic data

With $M = 200$, $N = 100$ and $K = 3$, the support set of the K-sparse signal $\mathbf{s}_t$ is chosen randomly, with nonzero entries drawn independently and identically distributed (i.i.d) from a standard Gaussian distribution. Matrix $\mathbf{D} \in \mathbb{R}^{M\times N}$ is randomly generated with entries sampled i.i.d. from the standard Gaussian distribution, and each column is normalized to have unit norm, while $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, 10^{-4}\mathbf{I})$. The algorithms are initialized with $\mathbf{D}[0] = \mathbf{D} + \mathbf{E}$, with $[\mathbf{E}]_{ij} \sim \mathcal{N}(0, 2.5 \times 10^{-2})$. Furthermore, T is allowed to vary between 50 and 300.

Setting the stepsize to a constant $\mu_t = 0.1$ in Algorithm 2, the developed binary dictionary learning (BDL) algorithms

("Online-BDL" and "Batch BDL") are compared with the ℓ_2-norm regularized binary dictionary learning algorithm in [16], henceforth referred to as "BDL2". In order to demonstrate the benefit of data-driven dictionary learning, performance comparisons are also drawn with binary iterative hard thersholding (BIHT), without dictionary learning (henceforth referred to as "BIHT-W") [6].

Figure 1 plots the normalized mean-square error (NMSE) defined as $\text{NMSE} := 20\log\left(\left\|\frac{\mathbf{X}}{\|\mathbf{X}\|_F} - \frac{\hat{\mathbf{X}}}{\|\hat{\mathbf{X}}\|_F}\right\|\right)$ for the different algorithms. It is clear that learning the dictionary from the data yields improved recovery performance. In addition, the developed algorithms markedly outperform BDL2, while online-BDL exhibits comparable performance to its batch counterpart.

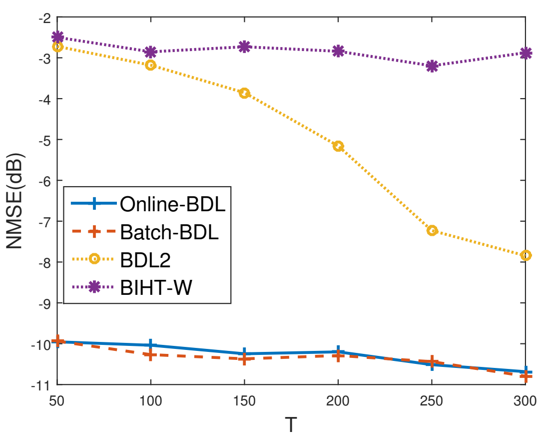

Fig. 1: Training NMSE (dB) vs. T.

Figure 2 depicts the training runtime in seconds against T, confirming that the developed batch algorithm is faster than BDL2 by at least an order of magnitude, thanks to the closed-form updates per iteration. Furthermore, online-BDL is faster than the batch-BDL as expected, which is quite appealing for processing streaming large-scale data.

Based on the computed $\hat{\mathbf{D}}$ from the training phase, reconstruction performance on 100 test data samples is also examined. The test data are generated using the same dictionary, under the same noise statistics, and only their signs are maintained. The sparse representations are then recovered via (15) based on $\hat{\mathbf{D}}$. From the NMSE performance illustrated in Figure 3, it is clear that even though BDL2 performs well on the training set, it does not perform well on the test set. In comparison, the developed algorithm still yields a consistent advantage in the test set too.

Finally, convergence performance of Online-BDL is tested. The norm of the gradient of the running average $\|(1/t)\sum_{\tau=1}^{t}\nabla_{\mathbf{D}} g_\tau(\{y_{i,t}\}_{i\in\Omega_t}; \mathbf{s}_t, \mathbf{D})\|_F$ is plotted against the iteration index as illustrated in Figure 4, which shows that the proposed online algorithm converges in about 50 iterations.

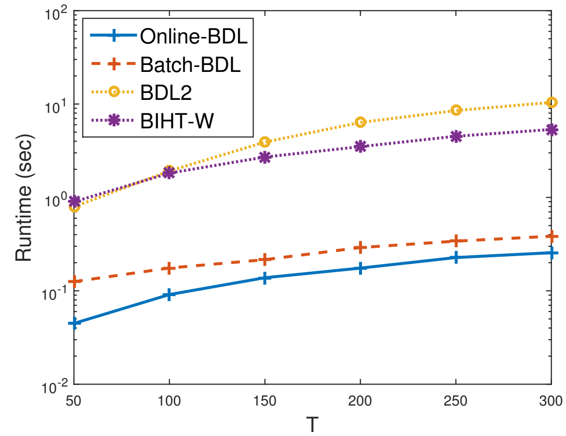

Fig. 2: Training time (sec) vs. T.

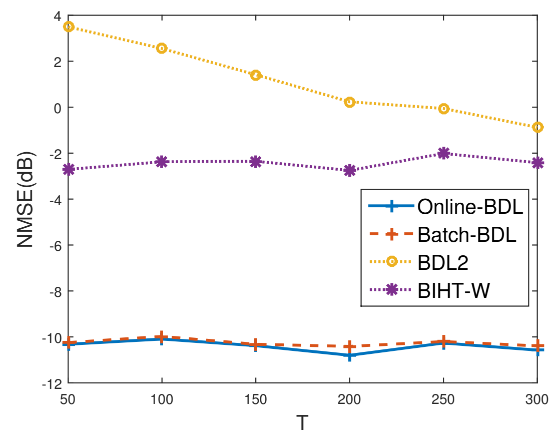

Fig. 3: Testing NMSE (dB) vs. T.

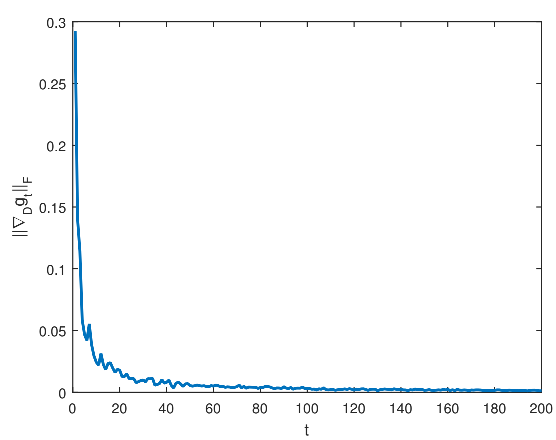

Fig. 4: Running average of gradient vs. t

B. MNIST data

The novel online algorithm is also evaluated on the MNIST dataset of handwritten digits. Four subsets of the data are randomly selected with $T = 100$ image patches corresponding to the numbers "1", "6","7", and "9." Each image of dimensions 28×28 is vectorized to form $\mathbf{s}_t \in \mathbb{R}^N$ with $N = 784$, and then multiplied by $\mathbf{D} \in \mathbb{R}^{5N \times N}$ generated as described in the previous subsection. The result is contaminated with additive Gaussian noise $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, 10^{-4}\mathbf{I})$. For BDL2, the maximum number of nonzero entries of $\mathbf{x}_t$ is known from the dataset. Table I lists the reconstruction performance in terms of NMSE, as well as the training runtime of the developed online approach and BDL2. To highlight the benefit of learning the dictionary, NMSE performance of "BIHT-W" is also included. Table I shows that the reconstruction performance improves through a data-driven approach for learning the dictionary. Furthermore, the developed online algorithm outperforms BDL2 on all four sets of data with respect to the NMSE metric.

With respect to scaling to large datasets, it is clear that the developed online algorithm outperforms BDL2, as demonstrated by the runtimes listed in Table I. Figure 5 visually compares the reconstruction performance of the different algorithms on images of the digit "6". Indeed, the developed approach yields a markedly improved reconstruction as seen from the higher visual quality in comparison to the rest.

	BDL-online		BDL2 [16]		BIHT-W
No.	runtime	NMSE	runtime	NMSE	NMSE
1	22.32	-11.08	506.93	-7.85	-7.68
6	21.51	-7.04	494.90	-6.33	-6.25
7	21.78	-7.39	495.46	-6.93	-6.88
9	22.12	-6.64	494.36	-6.23	-6.14

TABLE I: Runtime (sec) and NMSE (dB) comparison for MNIST dataset with $N = 784$, $M = 3920$, and $T = 100$.

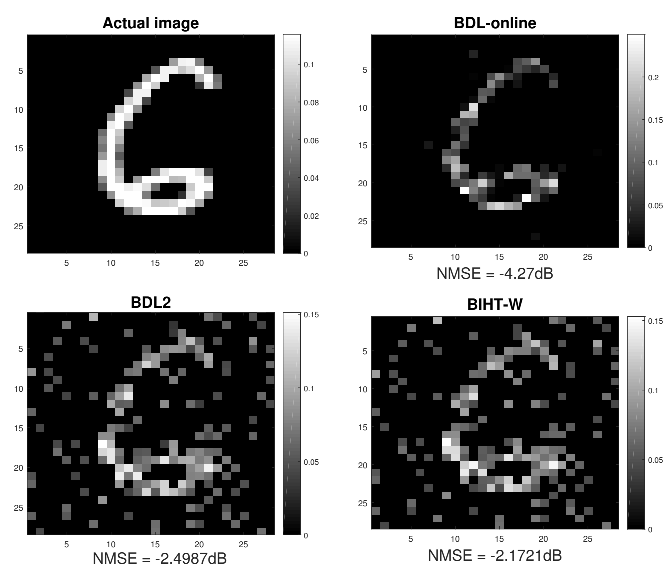

Fig. 5: MNIST image reconstruction test.

V. Conclusions

A novel provably convergent online approach was introduced for jointly recovering the sparse signal and learning the dictionary from large-scale binary data. Simulations on both synthetic and real data were carried out to corroborate the effectiveness of the proposed algorithm. To broaden the scope of this study, there are several intriguing directions to pursue, including: (a) convergence analysis of the iterative algorithm; and (b) leveraging kernels for dictionary learning in nonlinear settings.

Acknowledgement. The authors would like to thank Dr. B. Baingana for his input to this work.

References

[1] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[2] P. T. Boufounos and R. G. Baraniuk, "One-bit compressive sensing," in *Conf. on Info. Sciences and Systems*, Princeton, NJ, Mar. 2008.

[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Imag. Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[4] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Sig. Proc. Mag.*, vol. 25, no. 2, pp. 92–101, 2008.

[5] J. D. Haupt, N. Sidiropoulos, and G. B. Giannakis, "Sparse dictionary learning from 1-bit data," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 7664–7668.

[6] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *IEEE Trans. Info. Theory*, vol. 59, no. 4, pp. 2082–2102, Apr. 2013.

[7] Z. Kang and C. J. Spanos, "Sequential logistic principal component analysis (SLPCA): Dimensional reduction in streaming multivariate binary-state system," 2014. [Online]. Available: http://arxiv.org/abs/1407.4430

[8] S. Lee, J. Z. Huang, and J. Hu, "Sparse logistic principal components analysis for binary data," *Ann. of Appl. Stat.*, vol. 4, no. 3, pp. 1579–1601, 2010.

[9] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[10] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," May 2015, vol. 63, no. 10, pp. 2663–2677.

[11] A. H. Nuttall, "Some integrals involving the q-function," DTIC Document, Tech. Rep., 1972.

[12] M. Razaviyayn, M. Hong, and Z. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, pp. 1126–1153, Jun. 2013.

[13] Y. Shen, M. Mardani, and G. B. Giannakis, "Online sketching of big categorical data with absent features," in *Proc. of Conf. on Info. Sciences and Systems*, Baltimore, MD, Mar. 2015.

[14] K. Slavakis and G. B. Giannakis, "Online dictionary learning from big data using accelerated stochastic approximation algorithms," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 16–20.

[15] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[16] H. Zayyani, M. Korki, and F. Marvasti, "Dictionary learning for blind one bit compressed sensing," Aug. 2015. Available: http://arxiv.org/abs/1508.07648.

[17] L. Zhang, J. Yi, and R. Jin, "Efficient algorithms for robust one-bit compressive sensing," in *Proc. of Intern. Conf. on Mach. Learn.*, Beijing, China, May 2014, pp. 820–828.