ONE-BIT COMPRESSED SENSING USING GENERATIVE MODELS

Geethu Joseph, Swatantra Kafle, and Pramod K. Varshney

Department of EECS, Syracuse University, NY, USA
Emails: {gjoseph,skafle,varshney}@syr.edu

ABSTRACT

In this paper, we address the classical problem of one-bit compressed sensing. We present a deep learning based reconstruction algorithm that relies on a generative model. The generator which is a neural network, learns a mapping from a low dimensional space to a higher dimensional set comprising of sparse vectors. This pre-trained generator is used to reconstruct sparse vectors from their one-bit measurements by searching over the range of the generator. Hence, the algorithm presented in this paper provides excellent reconstruction accuracy by accounting for any other possible structure in the signal apart from sparsity. Further, we provide theoretical guarantees on the reconstruction accuracy of the presented algorithm. Using numerical results, we also demonstrate the efficacy of our algorithm compared to other existing algorithms.

***Index Terms*—** Sparsity, one-bit compressed sensing, deep learning, generative models

1. INTRODUCTION

The past couple of decades have seen an explosion of research activities in the area of compressed sensing (CS) [1, 2], and the popularity of CS has brought significant attention to quantized CS [3]. A well-studied version of quantized compressed sensing is called one-bit CS which is the extreme quantization setting where only the sign of the linear measurements is acquired. This set up possesses the advantages of low computational cost and easy implementation in hardware, but the one-bit measurements provide no information about the signal amplitude. Therefore, one-bit CS finds applications in systems that require the recovery of the unknown signal up to a scaling factor. For example, in a frequency division duplex massive MIMO system, the direction of the channel state information (CSI) at the transmitter is sufficient for the design of beam-forming vectors. In this case, using one-bit CS saves the uplink bandwidth resources required for the CSI feedback [4]. Some other fields where one-bit CS is used are radar [5], source localization [6], spectrum sensing [7], wireless sensor network [8], etc. Therefore, we focus on the one-bit CS problem, where the objective is to find an unknown sparse vector from one-bit its quantized linear measurements.

The one-bit CS was originally introduced in [9], and several sparse signal recovery algorithms are available in the literature such as convex optimization based algorithms [10], binary iterative hard thresholding [11], generalized approximate message passing based algorithm [12], binary sparse Bayesian algorithm [13], etc. These algorithms fall into the category of "traditional" algorithms as they are model-driven. Hence, the recovery accuracy depends on how efficiently the sparse structure is modeled by the algorithm. On the contrary, there is another class of algorithms that use a deep learning (DL) based approach and employ multiple layers of several non-linear transformations to progressively extract the underlying structure of the data from training samples. In this paper, we explore this possibility of using DL for one-bit CS.

Our algorithm uses a promising DL approach known as generative model based learning. The DL approach is already known to be useful for the recovery of a (non-sprase) vector from the one-bit measurements [14]. Our work is also motivated by the success of generative models like variational auto-encoders [15] and generative adversarial networks [16] at modeling data distributions. Our contributions are:

Algorithm development: We present a generative model based algorithm whose generative part learns the distribution of the sample space of sparse vectors. We then use gradient descent to optimize the representation learned by the model, that matches the given measurements.

Theoretical results: We establish that, as long as gradient descent finds a good approximate solution to our optimization problem, the algorithm output will be almost as close to the true vector as the closest possible point in the range of the generator.

Empirical validation: Through numerical simulations, we show that the presented algorithm requires fewer measurements than traditional one-bit CS algorithms for the same level of accuracy.

Overall, our algorithm is useful in learning a sparse vector from the signs of a small number of linear measurements.

2. SYSTEM MODEL AND ALGORITHM

We consider the problem of recovering an unknown sparse vector $\boldsymbol{x}^* \in \mathbb{R}^N$ from a set of one-bit measurements $\boldsymbol{y} \in$

This work was supported in part by NSF under Grant ENG 60064237.

$\{\pm 1\}^m$. The i^{th} measurement $\boldsymbol{y}_i$ is modeled as follows:

$$\boldsymbol{y}_i = \operatorname{sign}\left(\langle \boldsymbol{A}_i, \boldsymbol{x}^* \rangle\right) \in \{\pm 1\}, i = 1, 2, \ldots, m \qquad (1)$$

where $\boldsymbol{A}_i \in \mathbb{R}^N$ is the i^{th} row of the known measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times N}$. Our goal is to find a reconstruction $\hat{\boldsymbol{x}}$ such that it is close to the ground truth $\boldsymbol{x}^*$.

Our approach is to use a generative model which is given by a deterministic function: $G : \mathbb{R}^s \to \mathbb{R}^n$ and a distribution p_Z over $\mathbb{R}^s$. Typically, $s \ll n$, i.e., G is a mapping from a low dimensional representation space ($\subset \mathbb{R}^s$) to a high dimensional sample space ($\subset \mathbb{R}^n$), learned by the model. In the training phase, the algorithm learns the function G that can map the distribution p_Z to the distribution of the data from training samples. To be specific, we train the generative model using sparse vectors so that the range of the generator mapping is close to the desired set of sparse vectors.

In the testing phase, to reconstruct $\boldsymbol{x}^*$, we minimize the following objective function:

$$\operatorname{loss}(\boldsymbol{z}) = \|G(\boldsymbol{z})\|^2 - \frac{\sqrt{2\pi}}{m} \boldsymbol{y}^\top \boldsymbol{A} G(\boldsymbol{z}). \qquad (2)$$

The second term of the objective function maximizes the correlation between the one-bit measurements $\boldsymbol{y}$ and the corresponding linear measurements. For a fixed l_2 norm of $G(\boldsymbol{z})$, the term is maximized when $\operatorname{sign}(\boldsymbol{A}G(\boldsymbol{z})) = \boldsymbol{y}$. Therefore, the second term ensures the match between $\boldsymbol{A}G(\boldsymbol{z})$ and $\boldsymbol{y}$. However, the second term decreases as the l_2 norm of $G(\boldsymbol{z})$ increases, and therefore, we use the first term to control the norm. Hence, the two terms of the objective function jointly optimize the representation error.

Any optimization procedure can be used to minimize the loss function, and if the generative model is differentiable, we can use the standard back-propagation learning. Let $\hat{\boldsymbol{z}}$ denotes the optimization procedure output. Our reconstructed signal is given by $\hat{\boldsymbol{x}} = G(\hat{\boldsymbol{z}})$. We note that although the objective function to be optimized is non-convex, the optimization problem can be solved using gradient descent. We verify this point using empirical results presented in Section 4. Next, we discuss some theoretical guarantees for the above algorithm, assuming that the gradient descent finds a good approximate solution to the above non-convex optimization problem.

3. THEORETICAL ANALYSIS

Let G be a $d-$layer neural network with at most N nodes per layer, all weights are upper bounded by $w_{\max}$ in absolute value, and the non-linearity after each layer is $L-$Lipschitz. Further, let the range of the generator be denoted by $\mathcal{S}$. The main result of this section is as follows:

Theorem 1. *Let the input to the model G have independent entries drawn from a uniform distribution over* $\left[-\frac{r}{\sqrt{s}}, \frac{r}{\sqrt{s}}\right]$ *during the training phase, and* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *be a random Gaussian matrix, scaled so that* $\boldsymbol{A}_{i,j} \sim \mathcal{N}(0, 1/m)$. *Fix* $\boldsymbol{x}^*$ *satisfying* $\|\boldsymbol{x}^*\| = 1$. *Assume that the measurement vector* $\boldsymbol{y}$ *follows the model given by* (1). *Suppose* $\hat{\boldsymbol{z}}$ *minimizes the cost function in* (2) *to within additive* δ *of the optimum over the vectors with* $\|\boldsymbol{z}\| \leq r$. *Then, for any* $\epsilon > 0$ *there exists universal constants* $C, c > 0$ *such that if*

$$m \geq C\epsilon^{-2} s \left(r^2 + d \log LNw_{\max}\right), \qquad (3)$$

the following holds with probability at least $1 - 4\exp\left(-c\epsilon^2 m\right)$

$$\|G(\hat{\boldsymbol{z}}) - \boldsymbol{x}^*\|^2 \leq \min_{\substack{\boldsymbol{z} \in \mathbb{R}^s \\ \|\boldsymbol{z}\| \leq r}} \|G(\boldsymbol{z}) - \boldsymbol{x}^*\| + \delta + \epsilon. \qquad (4)$$

Proof. Let $\bar{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in \mathcal{S}} \|\boldsymbol{x}^* - \boldsymbol{x}\|$. At a high level, the main steps of the proof are as follows:

A . We first prove that, for any $\beta > 0$, the following holds with probability at least $1 - 4\exp\left(-2\beta^2\right)$,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 \leq \|\bar{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 + \delta + 4\sqrt{\frac{2\pi}{m}} \left(\mathscr{W}(\mathcal{S}) + \beta\right), \qquad (5)$$

where $\mathscr{W}(\mathcal{S})$ is the Gaussian mean width (See [17, Section 1.3] for the definition) of the range $\mathcal{S}$.

B . Next, we show that there exists a constant $C' > 0$ such that $\mathscr{W}(\mathcal{S})$ satisfies the following for any $r > 0$:

$$\mathscr{W}(\mathcal{S}) \leq 8r\sqrt{s} + C'\sqrt{sd\log(LNw_{\max})}. \qquad (6)$$

C . Finally, we combine the above two steps to bound $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|^2$ using an appropriately choice of β.

Step A: The proof of Step A is based on the concentration of the random function $f(\boldsymbol{x}) \triangleq \frac{1}{m}\sum_{i=1}^m \boldsymbol{y}_i \boldsymbol{A}_i^\top \boldsymbol{x}$ around its expectation, which is given in [17, Proposition 4.2]. By assumption, $\hat{\boldsymbol{x}} = G(\hat{\boldsymbol{z}})$ minimizes the cost function in (2) over $\mathcal{S}$ to within additive δ of the optimum. Thus, we get that

$$\begin{aligned}
-\delta &\leq \min_{\boldsymbol{x} \in \mathcal{S}} \left(\|\boldsymbol{x}\|^2 - \frac{\sqrt{2\pi}}{m}\boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{x}\right) - \left(\|\hat{\boldsymbol{x}}\|^2 - \frac{\sqrt{2\pi}}{m}\boldsymbol{y}^\top \boldsymbol{A}\hat{\boldsymbol{x}}\right) \\
&\leq \left(\|\bar{\boldsymbol{x}}\|^2 - \frac{\sqrt{2\pi}}{m}\boldsymbol{y}^\top \boldsymbol{A}\bar{\boldsymbol{x}}\right) - \left(\|\hat{\boldsymbol{x}}\|^2 - \frac{\sqrt{2\pi}}{m}\boldsymbol{y}^\top \boldsymbol{A}\hat{\boldsymbol{x}}\right) \\
&\leq \|\bar{\boldsymbol{x}}\|^2 - \|\hat{\boldsymbol{x}}\|^2 + \sqrt{2\pi} f(\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}) \qquad (7) \\
&\leq \|\bar{\boldsymbol{x}}\|^2 - \|\hat{\boldsymbol{x}}\|^2 + \sqrt{2\pi}\mathbb{E}\left\{f(\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}})\right\} \\
&\qquad + 4\sqrt{\frac{2\pi}{m}}\left(\mathscr{W}(\mathcal{S}) + \beta\right) \qquad (8) \\
&= \|\bar{\boldsymbol{x}}\|^2 - \|\hat{\boldsymbol{x}}\|^2 + 2\left(\hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}\right)^\top \boldsymbol{x}^* + 4\sqrt{\frac{2\pi}{m}}\left(\mathscr{W}(\mathcal{S}) + \beta\right) \\
&\qquad (9) \\
&= \|\bar{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 - \|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 + 4\sqrt{\frac{2\pi}{m}}\left(\mathscr{W}(\mathcal{S}) + \beta\right),
\end{aligned}$$

with probability at least $1 - 4\exp\left(-2\beta^2\right)$. Also, we use [17, Proposition 4.2] with parameter $t = \frac{4\beta}{\sqrt{m}}$, and [17, Lemma 4.1] to get (8) and (9), respectively. On rearranging the terms, we get the desired result.

Step B: The input to the generator $\boldsymbol{z}$ follows a uniform distribution and therefore, we get that $\|\boldsymbol{z}\| \leq r$. This in turn implies that $\mathcal{S} \subseteq G(\mathcal{B}_r^s)$,where $\mathcal{B}_r^s = \{\boldsymbol{z} \in \mathbb{R}^s : \|\boldsymbol{z}\| \leq r\}$ denote a ball of radius r. Next, we construct a $\frac{t}{(LNw_{\max})^d}-$cover $\mathcal{T}$ of $\mathcal{B}_r^s$ such that its cardinality is upper bounded by $\left(\frac{4r(LNw_{\max})^d}{t}\right)^s$ [18, Section 4.2.1]. Further, we use [19, Lemma 8.5.] to assert that G is $(LNw_{\max})^d-$Lipschitz. Hence, $G(\mathcal{T})$ is a $t-$cover of $G(\mathcal{B}_r^s)$, and thus, $G(\mathcal{T})$ is a $t-$cover of $\mathcal{S}$. Thus,

$$|G(\mathcal{T})| \leq |\mathcal{T}| \leq \left(\frac{4r(LNw_{\max})^d}{t}\right)^s \tag{10}$$

Therefore, for any $\boldsymbol{x} \in \mathcal{S}$, there exists a point $T(\boldsymbol{x}) = \underset{\boldsymbol{t} \in G(\mathcal{T})}{\arg\min} \|\boldsymbol{t} - \boldsymbol{x}\|$ such that $\|\boldsymbol{x} - T(\boldsymbol{x})\| \leq t$.

Having constructed a finite cover $G(\mathcal{T})$, we next bound the Gaussian mean width of $\mathcal{S}$. For any vector $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$,

$$\begin{aligned}
\mathscr{W}(\mathcal{S}) &\triangleq \mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \langle \boldsymbol{g}, (\boldsymbol{x}_1 - \boldsymbol{x}_2)\rangle\right\} && (11)\\
&\leq \mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \langle \boldsymbol{g}, (\boldsymbol{x}_1 - T(\boldsymbol{x}_1) + T(\boldsymbol{x}_2) - \boldsymbol{x}_2)\rangle\right\}\\
&\quad + \mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \langle \boldsymbol{g}, (T(\boldsymbol{x}_1) - T(\boldsymbol{x}_2))\rangle\right\}. && (12)
\end{aligned}$$

We further simplify the first term of the inequality as follows:

$$\begin{aligned}
&\mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \langle \boldsymbol{g}, (\boldsymbol{x}_1 - T(\boldsymbol{x}_1) + T(\boldsymbol{x}_2) - \boldsymbol{x}_2)\rangle\right\}\\
&\quad \leq \mathbb{E}\{\|\boldsymbol{g}\|\} \sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \|\boldsymbol{x}_1 - T(\boldsymbol{x}_1) + T(\boldsymbol{x}_2) - \boldsymbol{x}_2\| && (13)\\
&\quad \leq 2\sqrt{s} \sup_{\boldsymbol{x} \in \mathcal{S}} \|\boldsymbol{x} - T(\boldsymbol{x})\| \leq 2t\sqrt{s}. && (14)
\end{aligned}$$

Here, (13) follows from Cauchy-Schwarz inequality, and (14) uses the fact that $\mathbb{E}\{\|\boldsymbol{g}\|\} \leq \sqrt{\mathbb{E}\left\{\|\boldsymbol{g}\|^2\right\}} = \sqrt{s}$. Similarly, simplifying the second term of (12),

$$\begin{aligned}
&\mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}} \langle \boldsymbol{g}, (T(\boldsymbol{x}_1) - T(\boldsymbol{x}_2))\rangle\right\}\\
&\qquad \leq \mathbb{E}\left\{\sup_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in G(\mathcal{B}_r^s)} \langle \boldsymbol{g}, (\boldsymbol{x}_1 - \boldsymbol{x}_2)\rangle\right\} && (15)\\
&\qquad \leq \mathscr{W}(G(\mathcal{T})) && (16)\\
&\qquad = C' \sqrt{2s \log\left(\frac{4r(LNw_{\max})^d}{t}\right)}. && (17)
\end{aligned}$$

Here, (16) follows because $T(\boldsymbol{x}_1), T(\boldsymbol{x}_2) \in G(\mathcal{B}_r^s)$, and thus, supremum in (16) is over a larger set. Also, (17) follows from (10) and [17, Section 2.1]. Finally, using (12), (14), (17), we get the following:

$$\mathscr{W}(\mathcal{S}) \leq 2t\sqrt{s} + C' \sqrt{2s \log\left(\frac{4r(LNw_{\max})^d}{t}\right)}. \tag{18}$$

Finally, we choose $t = 4r$ to complete Step B.

Step C: Combining Steps A and B, we get that with probability at least $1 - 4\exp(-2\beta^2)$

$$\begin{aligned}
&\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 \leq \|\bar{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 + \delta\\
&\qquad + 4\sqrt{\frac{2\pi}{m}}\left(8r\sqrt{s} + C'\sqrt{sd \log LNw_{\max}} + \beta\right). && (19)
\end{aligned}$$

As given in the statement of the theorem, let the following lower bound on m holds for $C_1 > 64\pi$,

$$\begin{aligned}
m &\geq C_1 \epsilon^{-2} s \left(8r^2 + C' d \log LNw_{\max}\right) && (20)\\
&\geq \frac{C_1}{2\epsilon^2}\left(8r\sqrt{s} + C'\sqrt{sd \log LNw_{\max}}\right)^2 && (21)
\end{aligned}$$

If we choose $\beta = C_2 \epsilon \sqrt{m}$ with $C_2 = \frac{1}{4\sqrt{2\pi}} - \sqrt{\frac{2}{C_1}} > 0$,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 \leq \|\bar{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 + \epsilon + \delta. \tag{22}$$

with probability at least $1 - 4\exp\left(-c\epsilon^2 m\right)$. Finally, we also have

$$\|\bar{\boldsymbol{x}} - \boldsymbol{x}^*\|^2 = \min_{\substack{\boldsymbol{z} \in \mathbb{R}^s \\ \|\boldsymbol{z}\|_\infty \leq r}} \|G(\boldsymbol{z}) - \boldsymbol{x}^*\| \leq \min_{\substack{\boldsymbol{z} \in \mathbb{R}^s \\ \|\boldsymbol{z}\| \leq r}} \|G(\boldsymbol{z}) - \boldsymbol{x}^*\|.$$

Thus, the proof is complete. □

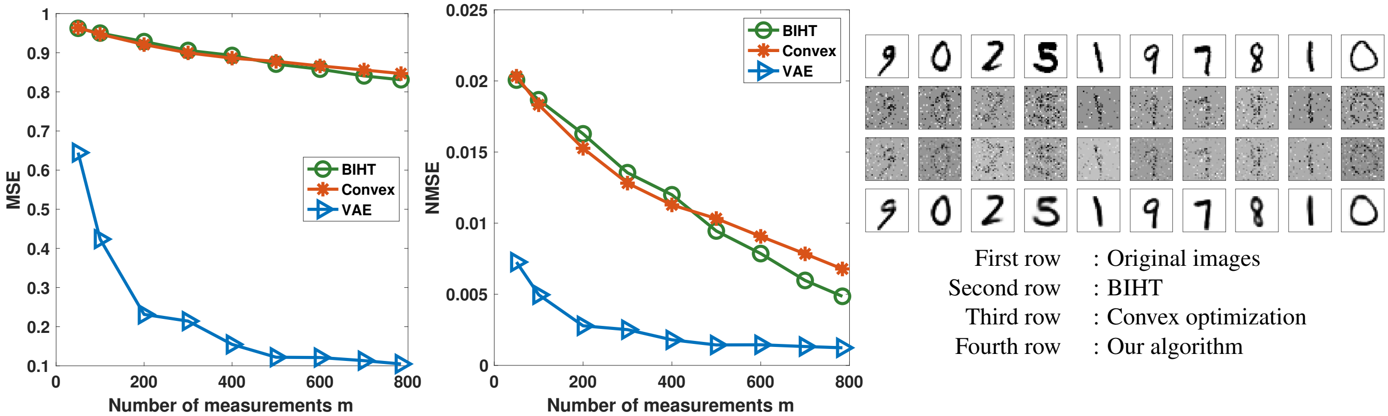
Discussion

Optimality of the error bound: The first term of the error bound arises because we search over the range of the generator to find the unknown sparse vector. The second term δ accounts for the fact that the gradient descent does not necessarily converges to the global optimum. Empirically, we see that these error terms converge to zero. These error terms are the minimum possible error terms for the presented technique, and therefore, they seem to be optimal for our algorithm.

Dependence on r: We see that as r increases, the number of measurements increases and the estimation error decreases. This is intuitive as r increases, the domain of the generator expands and thus, we need more measurements to train the generator. Further, as the domain expands, the range also becomes larger which results in improved accuracy.

Dependence on network parameters: As the network parameters, N, L and $w_{\max}$ increases, the number of measurements increases. This is because, as these parameters increases, the number of unknowns to be learned by the network or their respective ranges increase, and therefore, more number of measurements are required. On the other hand, this also increases the flexibility of network and thus, the range of $G(\boldsymbol{z})$ can become larger, and therefore, the first term in the error bound decreases. Hence, an increase in the number of measurements results in an improved error bound, as expected.

Distribution of input: Although the statement of the theorem specifies the distribution of the input to the generator as a uniform distribution parametrized by r, the proof only requires to assume that the norm of the input is bounded. Therefore, the



(a) MSE performance

(b) NMSE performance

(c) Reconstruction when $m = 300$.

Fig. 1. Reconstruction performance of our algorithm compared with the state-of-the-art algorithm.

result applies to an input $\boldsymbol{z}$ drawn from independent Gaussian or any other heavy-tailed distributions as this only prunes an exponentially unlikely fraction of the possible outputs.

4. SIMULATION RESULTS

In this section, the performance of our algorithm is evaluated using the MNIST handwritten digit dataset [20]. The dataset consists images whose pixel intensities are sparse (See the first row of Figure 1c). The experimental setup is as follows:

Generative model: We adopt the setup from [19] and train a variational autoencoder (VAE) [15] as the generative model using 60,000 training images from the MNIST dataset. The image size is 28 × 28, and thus, the input dimension is $N = 784$. We choose the size of the input to the generator as $s = 40$. The generator is a fully connected 784-500-500-40 neural network and the recognition network is a fully connected 40-500-500-746 network. The Adam optimizer [21] with a learning rate of 0.001 is used to train the VAE.

Measurement model: The columns of measurement matrix $\boldsymbol{A}$ are drawn uniformly from the surface of the m-dimensional unit hypersphere [22]. We use noisy compressed measurements $\boldsymbol{y}_i = \operatorname{sign}\left(\langle \boldsymbol{A}_i \boldsymbol{x}^* + \boldsymbol{n}_i \rangle\right) \in \{\pm 1\}, i = 1, 2, \ldots, m$, where the noise term $\boldsymbol{n}_i \sim \mathcal{N}(\boldsymbol{0}, 0.01\boldsymbol{I})$.

Performance metrics: We use two metrics: mean square error (MSE) $\triangleq \|\boldsymbol{x}^* - \hat{\boldsymbol{x}}\|^2$, and normalized mean square error (NMSE) $\triangleq \left\| \frac{\boldsymbol{x}^*}{\|\boldsymbol{x}^*\|} - \frac{\hat{\boldsymbol{x}}}{\|\hat{\boldsymbol{x}}\|} \right\|^2$. Here, $\boldsymbol{x}^*$ and $\hat{\boldsymbol{x}}$ denote the ground truth and the recovered vector, respectively. The both metrics are averaged over 20 Monte Carlo runs and 10 different images that were not included in the training.

Figure 1 compares the recovery performance of our algorithm (labeled as `VAE`) with the two traditional one-bit CS algorithms: convex optimization based algorithm in [17] (labeled as `Convex`) and binary iterative hard thresholding [11] (labeled as `BIHT`). The performance of all the algorithms improves with the number of measurements m, as more information about the sparse vector is available. However, our algorithm requires one order less number of measurements than the other algorithms for the same error. This is because the generative models, such as VAE that we used, act as a strong prior to the compressed signals, and hence, they provide better reconstruction performance. The MSE performance shown in Figure 1a is not surprising as our algorithm takes advantage of the training phase to give a better estimate of the magnitude of the signal. This, in turn, helps our algorithm to achieve lower MSE. Therefore, our algorithm can reconstruct the signal outside the unit ball if the signal lies in the range space of the generator. This is an added advantage of our algorithm compared to the traditional algorithms, which is due to the incorporation of a training phase. However, Figure 1b shows that our algorithm also provides better recovery of the direction of the unknown vector. Overall, our algorithm has better reconstruction accuracy compared to traditional algorithms, both in terms of the magnitude and the direction of the sparse vector. Figure 1c compares the images reconstructed by the above three algorithms with 300 measurements. The better visual quality of the images reconstructed by our algorithm demonstrates its superior performance compared to the competing algorithms.

5. CONCLUSIONS

We presented a one-bit compressed sensing algorithm using generative models. Unlike the prior work on this topic, our approach learns the underlying structure of the signal without explicitly depending on any sparsity model. We also provide reconstruction guarantees for the algorithm by characterizing the number of measurements that can achieve a given estimation error. Further, we empirically showed that our algorithm significantly outperforms the state-of-the-art algorithms.

6. REFERENCES

[1] M. Rani, S. Dhok, and R. Deshmukh, “A systematic review of compressive sensing: Concepts, implementations and applications,” *IEEE Access*, vol. 6, pp. 4875–4894, Jan. 2018.

[2] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee, “Compressive sensing: From theory to applications, a survey,” *J. Commun. Networks*, vol. 15, no. 5, pp. 443–456, Nov. 2013.

[3] Z. Li, W. Xu, X. Zhang, and J. Lin, “A survey on one-bit compressed sensing: Theory and applications,” *Front. Comput. Sci.*, vol. 12, no. 2, pp. 217–230, Apr. 2018.

[4] W. Tang, W. Xu, X. Zhang, and J. Lin, “A low-cost channel feedback scheme in mmwave massive mimo system,” in *Proc. ICC*, Dec. 2017, pp. 89–93.

[5] A. Ameri, A. Bose, J. Li, and M. Soltanalian, “One-bit radar processing with time-varying sampling thresholds,” *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5297–5308, Sep. 2019.

[6] Y. Shen, J. Fang, and H. Li, “One-bit compressive sensing and source localization in wireless sensor networks,” in *Proc. ChinaSIP*, Jul. 2013, pp. 379–383.

[7] S. B. J. Jian-Jun, “Investigation of sign spectrum sensing method [j],” *Acta Physica Sinica*, vol. 11, no. 32, p. 110701, 2011.

[8] D.-Y. Cao, K. Yu, S.-G. Zhuo, Y.-H. Hu, and Z. Wang, “On the implementation of compressive sensing on wireless sensor network,” in *Proc. IoTDI*, Apr. 2016, pp. 229–234.

[9] P. T. Boufounos and R. G. Baraniuk, “1-bit compressive sensing,” in *Proc. Conf. on Inform. Sci. and Syst. (CISS)*, 2008, pp. 16–21.

[10] Y. Plan and R. Vershynin, “One-bit compressed sensing by linear programming,” *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1275–1297, Aug. 2013.

[11] 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. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, Jan. 2013.

[12] J. Mo, P. Schniter, N. G. Prelcic, and R. W. Heath, “Channel estimation in millimeter wave mimo systems with one-bit quantization,” in *Proc. ASILOMAR*, Nov. 2014, pp. 957–961.

[13] J. Zhu, L. Han, X. Meng, and Z. Xu, “Binary sparse bayesian learning algorithm for one-bit compressed sensing,” *ArXiv e-prints*, May 2018. [Online]. Available: https://arxiv.org/abs/1805.03043

[14] S. Khobahi, N. Naimipour, M. Soltanalian, and Y. C. Eldar, “Deep signal recovery with one-bit quantization,” in *Proc. ICASSP*, May 2019, pp. 2987–2991.

[15] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, Dec. 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. NIPS*, Dec. 2014, pp. 2672–2680.

[17] Y. Plan and R. Vershynin, “Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach,” *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 482–494, Sep. 2012.

[18] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, 2018.

[19] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” in *Proc. ICML*, Aug. 2017, pp. 537–546.

[20] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” http://yann.lecun.com/exdb/mnist/, 2010.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[22] M. E. Muller, “A note on a method for generating points uniformly on n-dimensional spheres,” *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.