COMPRESSED SENSING OF GENERATIVE SPARSE-LATENT (GSL) SIGNALS

Antoine Honoré, Anubhab Ghosh, Saikat Chatterjee

Digital Futures and KTH Royal Institute of Technology, Stockholm, Sweden
honore@kth.se, anubhabg@kth.se, sach@kth.se

ABSTRACT

We consider reconstruction of an ambient signal in a compressed sensing (CS) setup where the ambient signal has a neural network based generative model. The generative model has a sparse-latent input and we refer to the generated ambient signal as generative sparse-latent signal (GSL). The proposed sparsity inducing reconstruction algorithm is inherently non-convex, and we show that a gradient based search provides a good reconstruction performance. We evaluate our proposed algorithm using simulated data.

Index Terms— Compressed sensing, generative models, inverse problems

1. INTRODUCTION

Background: Reconstruction of signals in an under-determined linear measurement setup, where the signals have generative models that are linear with latent sparse signals, has been thoroughly investigated in literature. This is primarily known as compressed sensing (CS) or compressed sampling [1, 2].

In this article we refer to the linear measurement setup alongwith the linear generative model as the traditional CS. Many practical algorithms exist for signal reconstruction using sparse-latent signals as a-priori knowledge. Algorithms are mainly divided as convex, greedy pursuits and Bayesian, and their suitable combinations [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Convex algorithms, such as LASSO [13], use ℓ_1-norm based penalties to promote sparsity.

Motivation: In this article, we consider a CS setup where we have linear measurements of an ambient signal like traditional CS, but do not have the linear generative model for the ambient signal. Instead, we assume non-linear, non-convex, generative models where an ambient signal is the output of a neural network excited by a sparse-latent input signal. Signals which are generated in such a way are referred to as '*generative sparse-latent*' (GSL) signals. Our objective is to design reconstruction algorithms for GSL signals in the CS setup.

In the CS setup measuring GSL, we have a non-linear mapping between the measurement and the sparse-latent signal. Due to the non-linear mapping, the reconstruction problem is inherently non-convex. We design a reconstruction algorithm that can use a-priori knowledge of sparsity. The reconstruction algorithm uses gradient search for sparse-latent estimation. Using simulations we show that it is possible to achieve a good quality signal reconstruction, although the mapping between measurement vector and sparse-latent signal is highly non-linear.

Naturally a question arises: how to measure a degree of non-linearity? Such a measure could help us to perform simulation studies in a controlled manner. To the best of authors' knowledge, we are not aware of any such measure. Therefore we provide a simple measure based on how far a non-linear mapping is from a linear mapping. This helps us to perform simulation studies in a controlled manner.

Relevant literature: In a CS setup, there exists prior work where neural networks-based generative models are used. In [14], variational auto-encoders (VAEs) and generative adversarial networks (GANs) are used where the dimension of latent signal is (much) lower than the dimension of ambient signal. In [15], the authors considered GAN-based generative models and consider the case where the dimension of the latent is small. Robustness of the CS for generative models was addressed in [16]. The work [17] considered normalizing flows (NFs) [18] that are invertible neural networks. These works address CS of signals with generative models – VAEs, GANs and NFs – excited with dense, Gaussian latent signals.

Our contributions: Use of sparse-latent signals for generative models in CS setup is the major novelty of the article compared to previous works. The contributions are: (a) Proposing the construction of GSLs. (b) Defining a measure of non-linearity and using the measure to rank complex non-linear mapping functions to generate GSL signals. (c) Proposing sparsity inducing reconstruction algorithms for CS of GSL signals and demonstrating their performances.

2. METHODS

2.1. GSL signals and its CS setup

A GSL signal $\mathbf{x}$ has the following generative model for the ambient signal:

$$\mathbf{x} \triangleq \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z}) \in \mathbb{R}^m, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^M$ is sparse, $\mathbf{B} \in \mathbb{R}^{m \times M}$ is a matrix, and $\mathbf{f}_{\boldsymbol{\theta}}(.) : \mathbb{R}^M \to \mathbb{R}^m$ is a neural network. Neural networks have arbitrarily complex mapping functions and are differentiable. We assume that $\|\mathbf{z}\|_0 \triangleq K < m \leq M$, where $\|.\|_0$ denotes ℓ_0-norm. The GSL model (1) is a generalization over the traditional linear generative sparse representation model where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ is linear (or identity mapping).

In this article, we investigate the reconstruction of GSL signal $\mathbf{x}$ from linear CS measurements. We assume that the GSL signal model and its parameters $\boldsymbol{\theta}, \mathbf{B}$ are known. We do not consider the modeling issues and learning parameters of GSL signals. In a CS setup, using linear measurements, we have n-dimensional noisy measurements:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} = \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z}) + \mathbf{n} \in \mathbb{R}^n, \ n < m, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^m$ is the signal to be reconstructed, $\mathbf{n}$ is additive measurement noise, and $n < m$. Using the a-priori information $\mathbf{z}$ be sparse, we need to solve the under-determined setup. We first estimate $\mathbf{z}$ as $\hat{\mathbf{z}}$ and then reconstruct $\mathbf{x}$ as $\hat{\mathbf{x}} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\hat{\mathbf{z}})$. Our investigation

Correspondence: sach@kth.se

in this article is concentrated on designing practical algorithms for quality reconstructions.

2.2. Optimization problems and reconstruction algorithm

We first start with the GSL CS setup (2) where the measurement noise $\mathbf{n}$ is absent ($\mathbf{n} = \mathbf{0}$). In that case, a relevant optimization problem is

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_1 \text{ s.t. } \mathbf{y} = \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z}). \tag{3}$$

The above problem has a resemblance with celebrated basis pursuit (BP) algorithm for the traditional CS. The major point is that BP is convex (a linear program) while the optimization problem in (3) is non-convex. BP provides exact reconstruction, i.e. $\hat{\mathbf{z}} = \mathbf{z}$, under certain technical conditions. In our understanding the optimization problem (3) is difficult to solve due to the (hard) equality constraint and presence of the non-linear mapping of the optimization variable $\mathbf{z}$.

While the measurement noise is absent or present, we address CS of GSL using the following optimization problem

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left\{ \lambda_1 \|\mathbf{y} - \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})\|_2^2 + (1 - \lambda_1)\|\mathbf{z}\|_1 \right\}, \tag{4}$$

where $\lambda_1 \in [0, 1]$ is a regularization parameter. The above optimization problem bears resemblance to the celebrated LASSO or basis pursuit denoising (BPDN) [19]. While LASSO / BPDN is convex, the optimization problem (4) is non-convex. To solve the problem we use gradient search. The gradient search is

$$\hat{\mathbf{z}}_{k+1} = \hat{\mathbf{z}}_k - \eta \left. \frac{\partial \mathcal{L}}{\partial \mathbf{z}} \right|_{\mathbf{z} = \hat{\mathbf{z}}_k}, \tag{5}$$

where $\mathcal{L}(\mathbf{z}) = \lambda_1 \|\mathbf{y} - \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})\|_2^2 + (1 - \lambda_1)\|\mathbf{z}\|_1$ and η is a tunable learning rate parameter. We use ADAM optimizer [20] to realize the gradient search. It is expected that the success of gradient search will be highly dependent on the degree of non-linearity in $\mathbf{f}_{\boldsymbol{\theta}}(.)$, and the choices of hyper-parameters λ_1 and η.

On quest of theoretical analysis, several questions can arise. Example questions are: (a) How good is the gradient search for solving the optimization algorithm (4)? (b) Can we reach to a globally optimum point? (c) If not, then how far is the local optimum point from the global optimum point? These questions are non-trivial given the optimization problem is non-convex. We do not deliberate on such theoretical questions in this article. Instead we investigate whether we can achieve empirical success using simulations.

2.3. Competing optimization problems

Competing optimization problems may use ℓ_2-norm based penalties on the ambient signal $\mathbf{x}$ or the latent signal $\mathbf{z}$. Without any signal structure imposition, a standard optimization problem is

$$\min_{\mathbf{x}} \left\{ \kappa \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + (1 - \kappa)\|\mathbf{x}\|_2^2 \right\}, \tag{6}$$

where $\kappa \in [0, 1]$ is a tunable parameter. Note that the optimization problem is convex (it is a regularized least-squares).

To use signal structures, algorithms in [14, 17] use the following non-convex optimization problem:

$$\min_{\mathbf{z}} \left\{ \lambda_2 \|\mathbf{y} - \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})\|_2^2 + (1 - \lambda_2)\, \|\mathbf{z}\|_2^2 \right\}, \tag{7}$$

where $\lambda_2 \in [0, 1]$ is a regularization parameter. In [14], authors investigated use of GAN and decoders of auto-encoders as the generative mapping function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})$, where $\dim(\mathbf{z}) \ll \dim(\mathbf{x})$ and it can happen that $\dim(\mathbf{z}) < \dim(\mathbf{y})$. Therefore the problem (7) while non-convex, may not be fully under-sampled. On the other hand, the work of [17] used normalizing flows (NFs) as the generative mapping function, where $\dim(\mathbf{z}) = \dim(\mathbf{x})$ due to invertibility requirement. Therefore $\dim(\mathbf{z}) > \dim(\mathbf{y})$ and the problem (7) is under-sampled in [17]. The optimization problem (7) can be considered state-of-the-art in CS for generative models.

2.4. A measure of non-linearity

In this subsection, we deliberate on how to measure non-linearity. We conjecture that the optimization problem (4) becomes hard to solve if $\mathbf{f}_{\boldsymbol{\theta}}$ becomes highly non-linear. This in turns raises the question: Does the optimization problem gets harder if $\mathbf{f}_{\boldsymbol{\theta}}$ gets more non-linear according to some metric of non-linearity? Then a natural queries can be: How to measure non-linearity? How to think that a function is more non-linear that another function? Can we conduct a systematic study to show that solving optimization problem (4) becomes harder with the increase in some measure of non-linearity?

We are not aware of a universal measure of non-linearity. Therefore we define a measure from a first principle - how much a non-linear mapping is away from an optimal linear mapping. The procedure is as follows.

For a given $\mathbf{f}_{\boldsymbol{\theta}}$, we first construct a dataset $\{(\mathbf{x}_j, \mathbf{z}_j)\}_{j=1}^{J}$ comprised of J data samples, where $(\mathbf{x}_j, \mathbf{z}_j)$ is j'th datum. We use the generative procedure $\mathbf{x} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})$ where $\mathbf{z}$ need not be sparse; $\mathbf{z}$ can be dense. Using this dataset, we construct the linear minimum-mean-square-estimation (LMMSE) as

$$\hat{\mathbf{x}}_L = C_{\mathbf{x}\mathbf{z}}(C_{\mathbf{z}\mathbf{z}} + \lambda I)^{-1}(\mathbf{z} - \mu_{\mathbf{z}}) + \mu_{\mathbf{x}}, \tag{8}$$

where $C_{\mathbf{x}\mathbf{z}}$ is the empirical cross-covariance between $\mathbf{x}$ and $\mathbf{z}$, $C_{\mathbf{z}\mathbf{z}}$ is the empirical covariance of $\mathbf{z}$, and $\mu_{\mathbf{z}}$ and $\mu_{\mathbf{x}}$ are the empirical means of $\mathbf{z}$ and $\mathbf{x}$, respectively, and $\lambda > 0$ is a tunable regularization parameter. The strength of non-linearities is quantified with a metric referred to as normalized-non-linearity-measure (NNLM) and defined as follows:

$$\text{NNLM}(\mathbf{f}_{\boldsymbol{\theta}}(.)) \quad = \frac{\mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}_L\|_2^2]}{\mathbb{E}[\|\mathbf{x}\|_2^2]} = \frac{\mathbb{E}[\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z}) - \hat{\mathbf{x}}_L\|_2^2]}{\mathbb{E}\big[\|\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})\|_2^2\big]}. \tag{9}$$

A high value of NNLM translates to a high non-linearity, and this should be associated with larger reconstruction errors. We choose λ appropriately using a cross-validation approach.

3. EXPERIMENTS

In this section we perform simulations using synthetic data. This helps our investigation in a controlled manner. The simulation were carried out in Python[1] using the PyTorch toolbox [21]. The experiments are carried out as follows.

1. In our first study, we use pretrained and untrained $\mathbf{f}_{\boldsymbol{\theta}}$ and study their NNLM score. This helps us to visualize how their degree of non-linearity varies.
2. The second study compares reconstruction performance of three competing algorithms for GSL CS, and show that the proposed algorithm (4) performs the best.

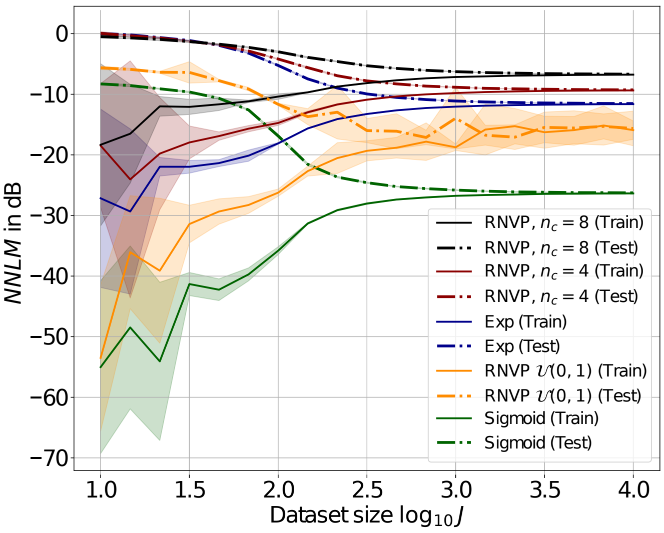

Fig. 1: Analysis of non-linearity

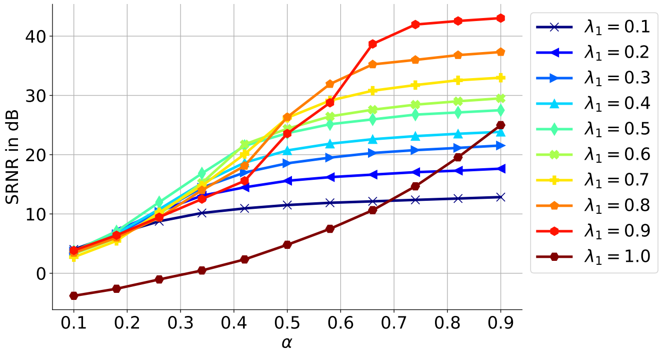

Fig. 2: SRNR performance of RNVP ($n_c = 4$) versus the sub-sampling ratio (α) for different values of λ_1.

3.1. First study - On non-linearity

In this study we consider several non-linear functions. (a) One-layer feedforward neural networks with either sigmoid or exponential activations. (b) RealNVP NF [22] with various layers. We conjecture that as the number of layers increases the mapping function becomes more non-linear. RealNVP (henceforth referred to as RNVP) is known to be a powerful generative model (NF), recently used in [17], and hence we choose it for our study. We use various configurations of RNVP.

In one configuration, a RNVP with 16 coupling layers is pretrained to map a isotropic Gaussian latent source to a uniform ambient distribution. The trained model is later used for our purpose where a GSL signal is generated by exciting the trained model using a sparse-latent signal. In other configurations we use untrained RNVP with 4 coupling layers and 8 coupling layers. In order to guarantee that the randomly initialized models are non-linear, the activation functions and the parameters priors for $\boldsymbol{\theta}$ are chosen according to [18]. Finally, the components of $\mathbf{B}$ are drawn from a normal distribution and the column are ℓ_2-normalized.

In the experiments, we use 100-dimensional isotropic Gaussian $\mathbf{z}$ to create signal $\mathbf{x}$, as mentioned in section 2.1. Linear estimators in high dimension are subject to the curse of dimensionality, making it possible to accurately estimate non-linear data with a linear transform on the training data set. We thus use the NNLM measure with a varying number of data points J, and by computing the NNLM on the training and a held-out test set.

The experimental results are shown in Fig. 1 for training data (used to learn the LMMSE parameters) and held out test data vis-a-vis number of data points J. In the figure, a higher NNLM value means a higher amount of non-linearity. From Fig. 1 we can observe that increase in the number of coupling layers from 4 (denoted as RNVP, $n_c = 4$) to 8 (denoted as RNVP, $n_c = 8$) for RNVP with random parameters leads to an increase in NNLM, which indicates a higher amount of non-linearity. The trained RNVP with 16 layers shows a lower NNLM than the RNVPs with random parameters, since it effectively learns the mapping of the Gaussian multivariate cumulative distribution function (by probability integral transform [23]). The trend of the curves is typical for training-testing scenarios. The ranking of the RNVP models according to an increasing trend of non-linearity is as follows: (i) RNVP $\mathcal{U}(0,1)$ as the pretrained RNVP, (ii) RNVP with random parameters and 4 coupling layers, (iii) RNVP with random parameters and 8 coupling layers.

3.2. Second study - Performance and comparison

For GSL CS, we now study the performances for three RNVP models: RNVP $\mathcal{U}(0,1)$, RNVP $n_c = 4$, and RNVP $n_c = 8$. The GSL signals are produced using sparse-latent $\mathbf{z}$.

We choose $m = M = 100$ and a sparsity level of 10%, i.e. $K = \|\mathbf{z}\|_0 = 0.1M = 10$. The support-set of $\mathbf{z}$ is chosen randomly and the non-zero components of $\mathbf{z}$ are drawn from a standard Gaussian. We add additive white Gaussian noise $\mathbf{n}$ in the signal-to-noise-ratio level 30 dB. We introduce sub-sampling ratio $\alpha = \frac{n}{m} \in [0.1, 0.9]$. We vary α and study reconstruction performance for GSL CS.

Following [8], we use two performance measures for the evaluation of the reconstruction: (1) signal to reconstruction noise ratio (SRNR) $= 10\log_{10}\frac{\mathbb{E}[\|\mathbf{x}\|_2^2]}{\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2]}$, and (2) average support cardinality error (ASCE) $= 1 - \frac{1}{K}\mathbb{E}\{|\mathcal{S}_{\mathbf{z}} \cap \mathcal{S}_{\hat{\mathbf{z}}}|\}$, where $\mathcal{S}_{\mathbf{z}}$ denotes the K-size support of $\mathbf{z}$. A K-size support set comprises of the indices of the K components of largest amplitude of $\mathbf{z}$.

We first study the effect of tuning λ_1 of (4). Fig. 2 shows SRNR performance of RNVP, $n_c = 4$, against α for different λ_1. We note that increase in λ_1 helps reconstruction and there exists a suitable range for an appropriate choice of λ_1. Note that $\lambda_1 = 1$ leads to no sparsity promoting penalty term and the performance degrades badly. The experiment confirms the importance of the sparsity-promoting penalty $\|\mathbf{z}\|_1$, and suggests that it is necessary to choose a suitable λ_1 to achieve good results.

Next we study the performances of three RNVP models and compare them. We include ℓ_2-penalty based optimization algorithm (7) where λ_2 is chosen appropriately. Fig. 3 shows SRNR and ASCE performances of methods. In the figure, λ_1 and λ_2 denotes the use of the optimization problem (4) and (7), respectively. It is clear that the use of sparsity promoting penalty $\|\mathbf{z}\|_1$ helps to achieve better reconstruction performances than the $\|\mathbf{z}\|_2^2$ penalty.

Another interesting result is that reconstruction algorithms show success according to the non-linearity level of $\mathbf{f}_{\boldsymbol{\theta}}$. Following our remarks at the end of section 3.1, we find that the performance curves in Fig. 3 are consistent with non-linearity level at the lower range of α. Lower range of α is of high importance in CS.

[1] The code is available upon request.

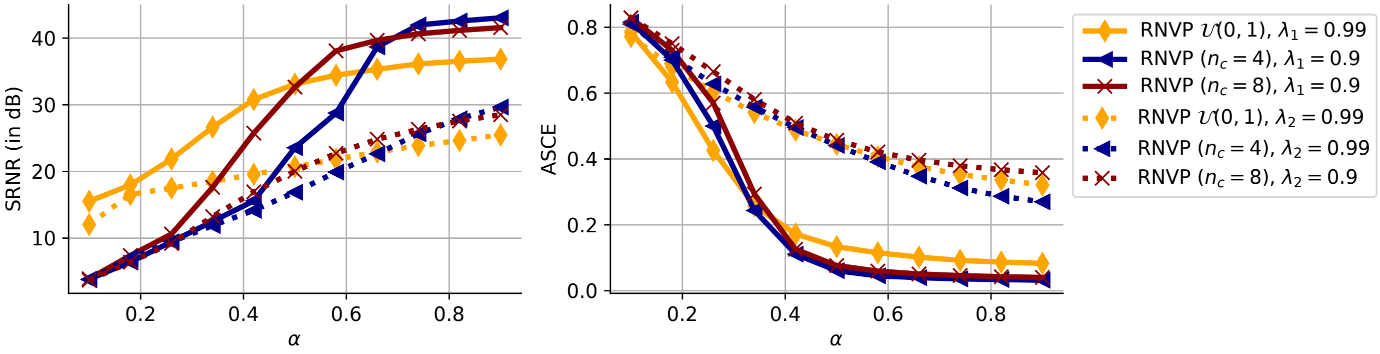

Fig. 3: Comparison of reconstruction methods in terms of SRNR and ASCE for 3 non-linearities.

We also investigated the regularized least-squares (6) and found that the performance are not promising. We skip those results to show in this article due to brevity.

Finally we show a GSL signal realization using RNVP, $n_c = 4$, and its reconstruction using the optimization algorithm (4). Fig. 4c shows a realization of GSL signal $\mathbf{x}$ and Fig. 4d shows its corresponding sparse-latent signal $\mathbf{z}$. This visualization helps to show that it is possible to generate ambient signals using sparse-latents. We also show reconstructed $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$. Fig. 4a demonstrates the gradient descent scheme in (5) and shows that the algorithm indeed converges to a local optimum. For the sake of completeness, we also show in Fig. 4b, the observation signal $\mathbf{y}$ and its reconstruction $\hat{\mathbf{y}}$ using $\hat{\mathbf{x}}$.

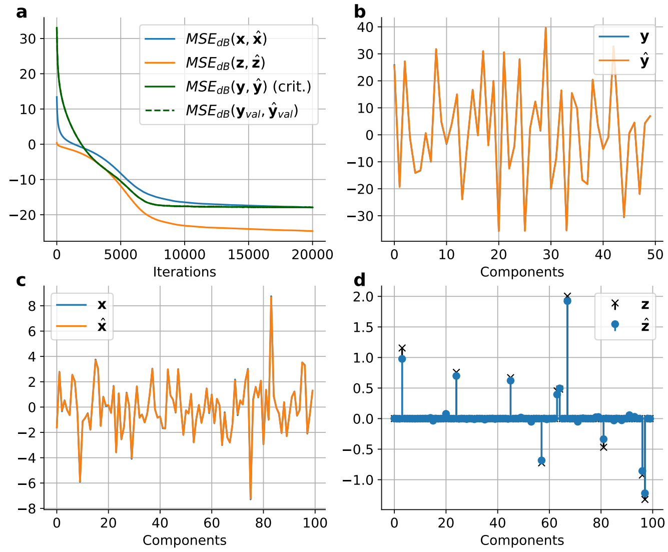

Fig. 4: Example of a successful reconstruction for one the J datum generated with a RNVP ($n_c = 4$) model and using ℓ_1 regularization with $\lambda_1 = 0.9$ and $\alpha = 0.5$.

3.3. Scope of improvements and challenges

While we mainly concentrate on the optimization problem (4) and its solution in this article using gradient descent, we delineate further improvement scopes and challenges.

Our first approach could be iteratively reweighted ℓ_1 minimization, where we solve

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left\{ \beta \|\mathbf{y} - \mathbf{A}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{B}\mathbf{z})\|_2^2 + (1-\beta)\|\mathbf{W}\mathbf{z}\|_1 \right\}, \qquad (10)$$

where β is a tunable parameter and $\beta \in [0, 1]$. The motivation of the improvement is from the work [9] applied to the traditional CS. Here $\mathbf{W}$ can be iteratively chosen to enhance sparsity in the same way of [9]. In the current iteration, we use $\mathbf{W}$ constructed from the solution of previous iteration. The same gradient-descent based approach of solving (4) can be used here in each iteration.

The next approach could follow the idea of iterative-reweighted-least-squares (IRLS) algorithm that has been in use for solving the traditional CS. In that case we write $\|\mathbf{z}\|_1$ in terms of ℓ_2-norm based representation, as $\mathbf{z}^\top[\text{diag}(\mathbf{1}./|\mathbf{z}|)]\mathbf{z}$, where $\mathbf{1}./|\mathbf{z}| = \left[\frac{1}{|z_1|} \; \frac{1}{|z_2|} \; \cdots \; \frac{1}{|z_M|}\right]$. In the current iteration we can use the solution of $\mathbf{z}$ from previous iteration to construct $\mathbf{1}./|\mathbf{z}|$. In each iteration, we can use the same approach of solving (7). Note that we did not experiment with the two possible scopes of improvements discussed above.

Finally, let us consider a Bayesian probabilistic approach. In that case, we wish to (1) find a posterior $p(\mathbf{z}|\mathbf{y})$ using a sparse prior $p(\mathbf{z})$, e.g. Laplacian prior or Gaussian-Gamma prior, and (2) find the posterior $p(\mathbf{x}|\mathbf{y})$ (rather than a point estimate). While relevance-vector-machine (RVM) and Bayesian CS with Gaussian-Gamma prior are used for the traditional CS [10, 24], an extension of such works for (2) perhaps comes with technical challenges. We believe the technical challenge will arise due to complicated non-linear functions $\mathbf{f}_{\boldsymbol{\theta}}$. This will have repercussions to compute $p(\mathbf{z}|\mathbf{y})$ and then translate the result to compute $p(\mathbf{x}|\mathbf{y})$.

4. CONCLUSION

In this study we introduced GSL signals and address their reconstruction in CS setup. We propose a measure of non-linearity for the generative models. We show that our measure is relevant to determine the quality of reconstruction at a low number of measurements. The proposed sparsity inducing reconstruction algorithm is shown to outperform competing methods in terms of the chosen performance measures – SRNR and ASCE. Future works include theoretical analysis and designing more efficient algorithms; also learning the parameters of GSL signal models for real signals.

5. ACKNOWLEDGEMENT

The authors thank Borja Rodríguez Gálvez (KTH) for fruitful discussions regarding the design of the non-linearity metric.

6. REFERENCES

[1] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289 –1306, April 2006.

[2] E.J. Candes and M.B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, pp. 21–30, march 2008.

[3] G. M. Davis, S. G. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Optical engineering*, vol. 33, no. 7, pp. 2183–2191, 1994.

[4] E.J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489 – 509, feb. 2006.

[5] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655 –4666, Dec. 2007.

[6] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *J. Found Comput. Math.*, vol. 9, pp. 317–334, 2009.

[7] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230 –2249, May 2009.

[8] S. Chatterjee, D. Sundman, M. Vehkaperä, and M. Skoglund, "Projection-based and look-ahead strategies for atom selection," *Signal Processing, IEEE Transactions on*, vol. 60, no. 2, pp. 634 –647, feb. 2012.

[9] E.J. Candes, M.B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *J. Fourier Anal. and Appl.*, vol. 14, no. 5, pp. 877–905, 2008.

[10] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Res.*, vol. 1, pp. 211–244, 2001.

[11] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346 –2356, June 2008.

[12] S.K. Ambat, S. Chatterjee, and K.V.S. Hari, "A committee machine approach for compressed sensing reconstruction," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1705–1717, 2014.

[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[14] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, Sydney, NSW, Australia, Aug. 2017, ICML'17, pp. 537–546, JMLR.org.

[15] Vinayak Killedar, Praveen Kumar Pokala, and Chandra Sekhar Seelamantula, "Sparsity driven latent space sampling for generative prior based compressive sensing," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2895–2899.

[16] A. Jalal, L. Liu, A. G. Dimakis, and C. Caramanis, "Robust compressed sensing using generative models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 713–727, 2020.

[17] X. Wei, H. van Gorp, L. Gonzalez-Carabarin, D. Freedman, Y. C. Eldar, and R. J. G. van Sloun, "Deep Unfolding With Normalizing Flow Priors for Inverse Problems," *IEEE Transactions on Signal Processing*, vol. 70, pp. 2962–2971, 2022.

[18] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2017, NIPS'17, pp. 972–981, Curran Associates Inc.

[19] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations (ICLR)*, 2015.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 721 in 1, pp. 8026–8037. Curran Associates Inc., Red Hook, NY, USA, Dec. 2019.

[22] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using RealNVP," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.

[23] J. E. Angus, "The probability integral transform and related results," *SIAM review*, vol. 36, no. 4, pp. 652–654, 1994.

[24] L. He and L. Carin, "Exploiting structure in Wavelet-Based Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488 –3497, sept. 2009.