Shape Parameter and Sparse Representation Recovery under Generalized Gaussian Noise

Florin Ilarion Miertoiu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
miertoiu.florin21@gmail.com

Bogdan Dumitrescu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
bogdan.dumitrescu@upb.ro

***Abstract*—We tackle the sparse representation problem with Generalized Gaussian noise from a new angle: we estimate the unknown shape parameter p of the noise distribution, not only compute the sparse representation. The procedure alternates between computing sparse representations for the current p and re-estimating p based on the empirical representation residual, until convergence. As basic sparse representation algorithm we propose a version of Feasibility Pump and show that it gives better results than ℓ_p versions of Orthogonal Matching Pursuit and ℓ_1 regularization. The results are comparable to those of the algorithms that know the true shape parameter value.**

***Index Terms*—mixed integer programming, sparse representation, feasibility pump, regularization, shape parameter estimation, probability density function**

I. Problem formulation

Sparse representations are an important mathematical tool that is useful in fields like machine learning, image processing or data classification. The sparse representation $\boldsymbol{x} \in \mathbb{R}^n$ of a signal $\boldsymbol{y} \in \mathbb{R}^m$ using the dictionary $\boldsymbol{D} \in \mathbb{R}^{m\times n}$, $m < n$, is the solution to the linear system $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$, where most of the coefficients of $\boldsymbol{x}$ are zero. In practice, noise affects the linear system and hence diverse optimization problems can be posed, depending on the type of noise and on the strategy to find the sparse representation.

A. Random noise model

Noise is omnipresent in signal processing applications. Usually, it is assumed that the noise is Gaussian, as this is the most common type of noise and it is also a robust choice in case the noise is unknown. Laplacian distribution is encountered for example in speech processing and better describes outliers. Impulsive noise is also met, particularly in image and video processing.

In most applications, an assumption is made on the noise distribution and the solution is given accordingly. However, in the situation where the noise structure is unknown, it may be more appropriate to work with a family of distributions and to identify the value of the parameters that characterize the distribution while solving the processing problem at hand. Our focus is on the Generalized Gaussian (GG) distribution. Our interest comes from the viewpoint of sparse representations.

The GG probability density function is [1]

$$g(\xi;\mu,\sigma,p) = \frac{1}{2\Gamma(1+1/p)A(p,\sigma)} e^{-\left|\frac{\xi-\mu}{A(p,\sigma)}\right|^p}, \xi \in \mathbb{R} \quad (1)$$

where μ represents the mean, σ^2 is the variance, p is the shape parameter and $A(p,\sigma)$ is a scaling factor; Γ is the Gamma function. The parameter p dictates the shape of the distribution; we name GG(p) the associated distribution (1). For $p = 1$ the distribution is Laplacian and for $p = 2$ the normal distribution is obtained.

Problems with GG(p) noise lead, through maximum likelihood, to optimization involving the p-norm. This is usually convenient for $p \geq 1$. However, for $p < 1$ the problems become non-convex and so harder to solve.

Our purpose in this paper is to estimate p while computing sparse representations, thus allowing better representations when the noise distribution is unknown.

B. Sparse representation problem

The sparse representation problem associated with GG(p) is

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_p \\ \text{subject to} \quad & \|\boldsymbol{x}\|_0 \leq K \end{aligned} \quad (2)$$

Here, K is the sparsity level, namely the number of atoms K that can be used for the representation $\boldsymbol{x}$. The problem can be formulated for $p > 0$, but we confine our study to $p \geq 1$, when the p-norm is truly a norm.

The problem (2) can be solved directly or can be relaxed by replacing the l_0 norm with the l_1 one, typically in the form

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_p + \lambda\|\boldsymbol{x}\|_1 \quad (3)$$

In most case, the shape parameter p is assumed to be known. However, using a different p can lead to larger approximation errors. Only if the shape parameter value is close to the true one, we can hope to get a solution $\boldsymbol{x}$ that is closer to the true solution. In this paper, we assume that p is unknown and attempt an explicit estimation of its value.

This work was supported in part by a grant of the Romanian Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-PED-2019-3248, within PNCDI III.

C. Existing work and contributions

A few examples of algorithms solving problems with GG(p) noise, not necessarily involving sparse representations, can be found in [2]–[4]. Sparse representations with GG(p) noise are computed in [5] using the ℓ_1 relaxation (3), and [6] with an adaptation of the Orthogonal Matching Pursuit (OMP) for solving (2).

There are also robust algorithms that do not make explicit assumptions on the noise distribution and also attempt to ignore outliers. Correntropy Matching Pursuit (CMP) [7] dynamically allocates weights in a weighted least squares scheme similar with OMP. A similar algorithmic approach, but with a different weighting mechanism was used in [8].

Mixed Integer Programming (MIP) algorithms can be used because the number of non-zero coefficients of $\boldsymbol{x}$ in (2) is always an integer. Similarly, MIP algorithms can be considered also for the binary decision whether a coefficient is part of the support of $\boldsymbol{x}$. In [9] and [10] a MIP algorithm was proposed called the Feasibility Pump. This algorithms aims to minimize the difference between the solution of the relaxed integer problem and the problem that satisfies the initial integer conditions. Several improvements have been proposed for this algorithm in [11]–[15]. For the ℓ_1 norm case, a modification of the Feasibility Pump algorithm for this problem was presented in [16], while the case which uses the ℓ_2 norm a modification was presented in [17].

In all the above works on sparse representations with GG(p) noise, it is assumed that the shape parameter p is known. We assume that p is *unknown* and our contribution is to propose a framework for estimating p and thus to compute the sparse representation with a value of p that is close to the true one. The procedure is iterative and simple, requiring only a sparse representation algorithm that works with fixed p.

The contents of the paper is as follows. Section II-A presents the shape parameter estimation and sparse representation framework. In section II-B we describe our Feasibility Pump algorithm adapted to GG(p) noise. Section II-C gives some details on other algorithms dedicated to the same problem that we have used for comparisons. Finally, section III presents experimental evidence showing that for low and medium values of the sparsity level, our framework is able to give accurate results on artificial data.

II. Algorithm

A. Shape parameter estimation framework

A key building block in our algorithm is the shape parameter estimation (SPE) method proposed in [1]. Given the errors $\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}$ of a representation (sparse and linear in our case, but the method is impervious to the model), a GG distribution is implicitly built to best approximate the empirical error distribution and so an estimation of the shape parameter value p is obtained. The method has very low complexity, of the order of the number of samples. More error samples lead to better estimation, so we consider the general case in which t signals are simultaneously represented; this is an assumption that is valid in many situations, especially in the dictionary learning context.

Data: Signal to represent $\boldsymbol{y} \in \mathbb{R}^{m \times t}$, dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, sparsity level $K \in \mathbb{Z}$, maximum number of iterations for ℓ_p norm estimation $Iter_{norm}$, stopping threshold θ
Result: Sparse representations $\boldsymbol{x} \in \mathbb{R}^{n \times t}$, estimated shape parameter $p \in \mathbb{R}$, $p \geq 1$

```
Compute sparse representations x using algorithm with p = 2 for each column of y
Use algorithm in [1] to estimate shape parameter p̃ from representation errors
Update norm p using (4)
while number of iterations ≤ Iter_norm or |p − p̃| > θ do
    Compute sparse representations x using algorithm with p from the previous step
    Use algorithm in [1] to estimate shape parameter p̃
    Update p using (4)
end
```

Algorithm 1: Algorithm for shape parameter estimation

In general, a sparse representation algorithm (SRA) has an underlying assumption of the noise characteristics. For example, OMP assumes a Gaussian distribution; as a result, the actual a posteriori error distribution is not far from a Gaussian one, no matter what the (unknown) actual noise distribution is.

Assuming that we have noise only within the GG family, with unknown shape parameter $p_{\text{true}} \geq 1$, there are also SRAs that have a certain flexibility (we use the acronym FSRA – Flexible SRA) and can work with any given p, like those in [5], [6]. The problem is that the shape parameter p_{true} is often not known a priori. So, in most of the cases, if we compute the error $\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}$ obtained by a FSRA and then apply the SPE algorithm [1], the resulting shape parameter $\tilde{p}$ is clearly different from p_{true} as well as from the p we have used.

Our purpose is the estimation of p_{true} using a FSRA. The motivation is immediate: a FSRA in possession of p_{true} is more likely to produce a better representation than a FSRA working with a different p or than a SRA that, like OMP and many others, works with fixed p. We propose algorithm 1 to compute an estimation of the shape parameter, together with the sparse representation.

The idea is to start with a given p (we start from 2, as in most cases the noise is Gaussian). At each step of our iterative algorithm, the error $\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}$ is computed and the SPE algorithm [1] is used to compute the associated $\tilde{p}$. Then, we update the norm via

$$p \leftarrow (p + \tilde{p})/2. \tag{4}$$

So, we go towards p_{true} as guided by the empirical noise distribution, but temper the change in p in order to prevent oscillations. The sparse representation is computed with the new p and so on.

Data: Signal to represent $\boldsymbol{y} \in \mathbb{R}^m$, dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, sparsity level $K \in \mathbb{Z}$, maximum number of iterations $Iter$, weights α, λ, γ

Result: Sparse representation $\boldsymbol{x} \in \mathbb{R}^n$

1 Solve relaxed (5) with $\boldsymbol{b} \in [0,1]^n$. The vectors $\boldsymbol{x}$ and $\boldsymbol{b}$ are obtained.
2 Use rounding procedure to obtain vector $\widetilde{\boldsymbol{b}}$.
3 **while** *number of iterations* $\leq Iter$ **do**
4   Solve problem (6) for $\boldsymbol{x}$ and $\boldsymbol{b}$.
5   **if** $\boldsymbol{b}$ *is integer* **then**
6   exit loop;
7   **end**
8   Use rounding procedure to obtain vector $\widetilde{\boldsymbol{b}}$
9   **if** *cycle is detected* **then**
10   Perturb $\widetilde{\boldsymbol{b}}$
11   **end**
12   Update $\alpha \leftarrow \gamma\alpha$
13 **end**
14 Return $\boldsymbol{x}$, optimized with Least Squares Method on the found support.

Algorithm 2: Modified Feasibility Pump

This framework is designed so that any FSRA can be used with it. The algorithm stops after the difference between $|p-\tilde{p}|$ is under a certain threshold θ. This threshold affects the speed of the algorithm because more iterations are necessary if the value θ is too small. Since it is unreasonable to aim for a very precise estimation of p_{true}, we keep θ large enough. Also, we limit the number of steps to $Iter_{norm}$, in order to stop possibly erratic behavior. The number of iterations should anyway be small, otherwise the algorithm becomes unpractical.

B. Sparse representation with ℓ_p norm using the Feasibility Pump

The Mixed Integer Programming (MIP) proposed in [18] introduces the binary variable $\boldsymbol{b} \in \{0,1\}^n$ to indicate which atom of the dictionary $\boldsymbol{D}$ is part of the support of $\boldsymbol{x}$. Applying this idea to (2) and combining with the Lasso problem (3) for faster sparsity enhancement, the following reformulation is obtained:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{b}\in\{0,1\}^n}{\text{minimize}} \quad & \|\boldsymbol{y}-\boldsymbol{D}\boldsymbol{x}\|_p + \lambda\|\boldsymbol{x}\|_1 \\ \text{subject to} \quad & \mathbf{1}_n^T\boldsymbol{b} \leq K \\ & -M\boldsymbol{b} \leq \boldsymbol{x} \leq M\boldsymbol{b} \end{aligned} \tag{5}$$

where $\mathbf{1}_n$ is a vector of length n whose elements are all equal to 1 and M is a large constant.

We propose Algorithm 2, which consists of a modification of the Feasability Pump (FP) and has a structure similar with that from [16] and [17].

The algorithm repeatedly solves two problems in which the binary variable $\boldsymbol{b}$ is relaxed to the interval $[0,1]^n$. The solution is then rounded to $\widetilde{\boldsymbol{b}}$, the nearest binary vector with K values of one. One of the problems is (5). The other is

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{b}\in[0,1]^n}{\text{minimize}} \quad & (1-\alpha)\triangle(\boldsymbol{b},\widetilde{\boldsymbol{b}}) + \alpha\left[\|\boldsymbol{y}-\boldsymbol{D}\boldsymbol{x}\|_p + \lambda\|\boldsymbol{x}\|_1\right] \\ \text{subject to} \quad & \mathbf{1}_n^T\boldsymbol{b} \leq K \\ & -M\boldsymbol{b} \leq \boldsymbol{x} \leq M\boldsymbol{b} \end{aligned} \tag{6}$$

where the term $\triangle(\boldsymbol{b},\widetilde{\boldsymbol{b}}) = \|\boldsymbol{b}-\widetilde{\boldsymbol{b}}\|_1$ aims to reduce the difference between the two solutions (relaxed and integer) [19]. The decay term α accelerates convergence; taking $0 < \gamma < 1$ in step 12 of the algorithm decreases the weight of the error term in favor of the nearness term.

Both problems, the relaxed (5) and (6), are convex and we solve them with CVX [20]; other solvers can be used.

The cycle detection and perturbation procedure is described in [17]. We also note that, besides the use of the p-norm, the objective of (6) differs from that in [17] by the lack of extra weighting to compensate for the difference in magnitude between terms.

C. Other algorithms for ℓ_p sparse representation

The greedy OMP-ℓ_p algorithm [6] for solving (2) is the direct generalization to the p-norm of the standard OMP [21]. More precisely, if $\bar{D}$ is the set of atoms selected by OMP-ℓ_p at the current step and $\boldsymbol{r} = \boldsymbol{y} - \bar{\boldsymbol{D}}\boldsymbol{x}$ the current residual, the next selected atom $\boldsymbol{d}$ is the one maximizing the projection

$$\min_{\xi} \|\boldsymbol{r} - \boldsymbol{d}\xi\|_p \tag{7}$$

The selected atom is appended to the set $\bar{\boldsymbol{D}}$ and the associated optimal representation $\boldsymbol{x}$ is computed from

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \bar{\boldsymbol{D}}\boldsymbol{x}\|_p \tag{8}$$

Both problems (7) and (8) are convex and the first has a single variable. Moreover, good initializations are available. In problem (7), the ℓ_2-optimal projection $\boldsymbol{r}^T\boldsymbol{d}$ can be used; also, solving (7) can be limited to the atoms for which the product $\boldsymbol{r}^T\boldsymbol{d}$ is high enough, since atoms that are nearly orthogonal cannot have a good correlation in the ℓ_p norm, whatever is the value of p; we have used a threshold equal to $0.5\max_{\boldsymbol{d}}|\boldsymbol{r}^T\boldsymbol{d}|$. In problem (8), we initialize the older elements of $\boldsymbol{x}$ with their values from the previous step and the new element with value ξ resulted from (7).

Several algorithms have been proposed for the solution of the relaxed ℓ_p sparse representation problem (3). Since the problem is convex, all algorithms give basically the same solution, the difference being especially in complexity. The main issue is in fact the choice of the parameter λ.

In [5] a modification of the alternating direction method (ADMM) that adds the proximal operator of l_p-norm functions to the framework of augmented Lagrangian methods is proposed. The proximal operators are used to estimate the solution of (3) depending on the interval in which p is.

Also in [22] another modification of ADMM is proposed in which the Continuous Mixed Norm proposed [23] is used as penalty function; in their practical implementation, it is replaced by a surrogate function.

III. RESULTS

The numerical results are obtained using a testing scheme similar to that from [16] and [17], with the significant distinction that the perturbation noise is now Generalized Gaussian.

Dictionaries of size 50×100 are generated randomly. For each dictionary, $t = 50$ test signals are generated via $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}_{\text{true}} + \boldsymbol{u}$, where the true solutions x_{true} are random and have sparsity level $K \in \{4, 6, 8\}$. The perturbation noise u is Generalized Gaussian and its variance is chosen such that the signal to noise ratio is 30. The shape parameters used for the noise are $p_{\text{true}} \in \{1, 1.2, 1.4, 1.6, 1.8, 2\}$. For each combination K, p, we test 10 different data sets.

We name FP-GGN the framework algorithm 1 in which the Feasibility Pump algorithm 2 is integrated. Its results are the sparse representation $\boldsymbol{x}$ and the shape parameter p. Similarly, for comparison purposes, we insert in algorithm 1 the ℓ_p sparse representation algorithms from [5] (named LP_L1) and [6] (named OMP-p).

We have also tested other algorithms: OMP, as a representative of the algorithms based on normal distribution of errors ($p = 2$); RLAD [24], as a representative of algorithms using a Laplacian distribution of errors ($p = 1$); and the CMP algorithm [7], as a member of the robust methods family. All three algorithms gave clearly worse results than the above adaptive algorithms for most (OMP, RLAD) or all (CMP) p_{true} values, so we will report no numerical results for them.

The algorithms were implemented in MATLAB, using the CVX library for FP-GGN, and tested on a computer with a 6-core 3.4 GHz processor and 32 GB of RAM.

The algorithms are compared in terms of mean representation errors, recovery errors and estimated shape parameters. The relative representation error

$$e_{\text{rep}} = \|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{y}\|_p \;/\; \|\boldsymbol{y}\|_p \tag{9}$$

is used (where now $\boldsymbol{x}$ is the computed solution), in accordance with the formulation of the basic problem (2). The relative recovery error is

$$e_{\text{rec}} = \|\boldsymbol{x} - \boldsymbol{x}_{\text{true}}\|_p \;/\; \|\boldsymbol{x}_{\text{true}}\|_p \tag{10}$$

Table I gives the mean over the 10 runs of the estimated shape parameters p, for the three algorithms under scrutiny. Figure 1 contains the full information for all runs with $K = 4$ (with stars), the mean being displayed with a different symbol; the horizontal displacement is used only for better visibility; the values p_{true} are the same for all algorithms. We note that the shape parameter is recovered well enough, especially for the lower values of the sparsity level K. The worst approximations appear when $p_{\text{true}} = 1$; a contributing cause is the fact that values $p < 1$ are not possible, hence a certain inherent bias.

The mean representation and recovery errors are shown in Tables II and III, respectively. For each K, there are two columns: the left one contains the errors of our adaptive algorithms for which p_{true} is unknown; the right one contains the errors of the same basic algorithms that are in possession of p_{true} and hence are run only once.

TABLE I
MEAN ESTIMATED SHAPE PARAMETERS FOR OMP-P (TOP IN EACH CELL), FP-GGN (MIDDLE) AND LP_L1 (BOTTOM)

K	4	6	8
p=1	1.1318 1.1236 1.0988	1.1854 1.1524 1.1809	1.2333 1.2592 1.1905
p=1.2	1.1979 1.2154 1.2164	1.1562 1.1865 1.1935	1.0924 1.0964 1.1174
p=1.4	1.3824 1.4025 1.4112	1.3317 1.3670 1.3913	1.2620 1.2477 1.2802
p=1.6	1.6396 1.6381 1.6343	1.5210 1.5692 1.5792	1.5365 1.4813 1.5559
p=1.8	1.8189 1.8156 1.8317	1.8346 1.8291 1.8453	1.8277 1.8034 1.8107
p=2	2.0148 2.0283 2.0171	2.0576 2.0399 1.9970	2.1439 2.0772 2.2541

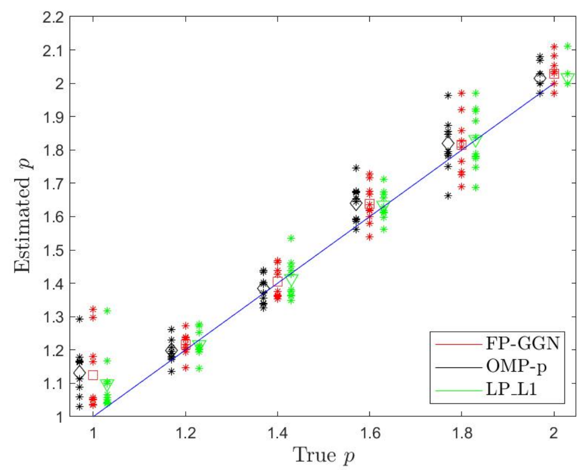

Fig. 1. True and estimated shape parameters for K=4

We note that: i) the errors of the adaptive algorithms are near from those of the algorithms knowing p_{true}, occasionally better; so, our proposed framework is able to provide good representations even though p_{true} is unknown; ii) FP-GGN offers better error recovery in most cases; the results of OMP-p are usually good, but large errors appear in few runs; LP_L1 is more reliable.

Regarding support recovery, FP-GGN misses 0.184 positions per test (true support recovered in 82.4% of cases, OMP-p misses 0.500 positions per test (true support recovered in 74.8% of cases) and LP_L1 misses 0.267 per test (true support recovered in 76.9% of cases). It can be seen that FP-GGN gets the correct support in more cases that the other two algorithms.

The running time of FP-GGN is 1870s per test, OMP-p takes 5.9s and LP_L1 takes 6.3s. So, using a MIP-based approach takes a larger amount of time. This is also caused by the CVX library, which is more suited to large problems.

TABLE II
MEAN REPRESENTATION ERRORS FOR OMP-P (TOP IN EACH CELL), FP-GGN (MIDDLE) AND LP_L1 (BOTTOM)

K	4		6		8	
	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed
p=1	0.0343	0.0359	0.0447	0.0495	0.0505	0.0611
	0.0271	0.0271	0.0262	0.0262	0.0255	0.0256
	0.0272	0.0272	0.0263	0.0262	0.0262	0.0261
p=1.2	0.0283	0.0283	0.0384	0.0383	0.0590	0.0611
	0.0286	0.0286	0.0281	0.0280	0.0273	0.0271
	0.0289	0.0289	0.0279	0.0279	0.0279	0.0278
p=1.4	0.0293	0.0293	0.0326	0.0309	0.0505	0.0467
	0.0296	0.0296	0.0290	0.0288	0.0282	0.0280
	0.0308	0.0308	0.0291	0.0290	0.0286	0.0285
p=1.6	0.0298	0.0298	0.0306	0.0310	0.0457	0.0449
	0.0300	0.0300	0.0293	0.0292	0.0287	0.0286
	0.0300	0.0300	0.0295	0.0294	0.0295	0.0292
p=1.8	0.0326	0.0326	0.0323	0.0332	0.0478	0.0475
	0.0302	0.0302	0.0295	0.0292	0.0288	0.0286
	0.0306	0.0304	0.0296	0.0295	0.0291	0.0289
p=2	0.0306	0.0306	0.0359	0.0350	0.0456	0.0442
	0.0303	0.0303	0.0296	0.0296	0.0289	0.0289
	0.0310	0.0310	0.0302	0.0301	0.0294	0.0294

TABLE III
MEAN RECOVERY ERRORS FOR OMP-P (TOP IN EACH CELL), FP-GGN (MIDDLE) AND LP_L1 (BOTTOM)

K	4		6		8	
	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed
p=1	0.0211	0.0235	0.0536	0.0636	0.0800	0.1106
	0.0089	0.0093	0.0113	0.0123	0.0131	0.0142
	0.0095	0.0097	0.0116	0.0119	0.0150	0.0154
p=1.2	0.0080	0.0079	0.0356	0.0362	0.1022	0.1107
	0.0088	0.0088	0.0121	0.0121	0.0159	0.0149
	0.0098	0.0098	0.0117	0.0117	0.0174	0.0171
p=1.4	0.0094	0.0093	0.0182	0.0152	0.0688	0.0617
	0.0101	0.0100	0.0122	0.0121	0.0159	0.0154
	0.0119	0.0119	0.0126	0.0126	0.0169	0.0168
p=1.6	0.0094	0.0093	0.0147	0.0152	0.0622	0.0606
	0.0096	0.0097	0.0129	0.0130	0.0153	0.0153
	0.0098	0.0098	0.0136	0.0134	0.0176	0.0168
p=1.8	0.0132	0.0132	0.0176	0.0191	0.0664	0.0694
	0.0095	0.0097	0.0126	0.0153	0.0162	0.0162
	0.0103	0.0101	0.0126	0.0124	0.0170	0.0166
p=2	0.0094	0.0094	0.0224	0.0202	0.0579	0.0566
	0.0093	0.0092	0.0119	0.0118	0.0162	0.0161
	0.0100	0.0104	0.0134	0.0133	0.0180	0.0179

IV. CONCLUSION

We have proposed a simple framework for sparse representation under Generalized Gaussian noise with unknown shape parameter value p. Any sparse representation algorithm that can work with known p can be used within this framework. Our tests show that the Feasibility Pump algorithm that we propose is better in terms of representation and recovery errors, although slower. Results are especially good at low sparsity level.

Future research will be dedicated to faster implementations and to the extension of our framework to dictionary learning.

REFERENCES

[1] J. A. Dominguez-Molina, G. González-Farías, R. M. Rodríguez-Dagnino, and I. C. Monterrey, "A practical procedure to estimate the shape parameter in the generalized gaussian distribution," 2003. Available at http://www.cimat.mx/reportes/enlinea/I-01-18_eng.pdf.

[2] O. Taheri and S. A. Vorobyov, "Sparse channel estimation with lp-norm and reweighted l1-norm penalized least mean squares," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2864–2867, IEEE, 2011.

[3] D. Adil, R. Kyng, R. Peng, and S. Sachdeva, "Iterative refinement for lp-norm regression," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1405–1424, SIAM, 2019.

[4] L. Leindler, "Trigonometric approximation in lp-norm," *Journal of Mathematical Analysis and Applications*, vol. 302, no. 1, pp. 129–136, 2005.

[5] F. Wen, P. Liu, Y. Liu, R. C. Qiu, and W. Yu, "Robust sparse recovery in impulsive noise via ℓ_p-ℓ_1 optimization," *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 105–118, 2016.

[6] W. Zeng, H. So, and X. Jiang, "Outlier-Robust Greedy Pursuit Algorithms in ℓ_p-Space for Sparse Approximation," *IEEE Trans. Signal Processing*, vol. 64, no. 1, pp. 60–75, 2016.

[7] Y. Wang, Y. Tang, and L. Li, "Correntropy matching pursuit with application to robust digit and face recognition," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1354–1366, 2016.

[8] C. Loza, "RobOMP: Robust variants of Orthogonal Matching Pursuit for sparse representations," *PeerJ Computer Science*, vol. 5, p. e192, 2019.

[9] M. Fischetti, F. Glover, and A. Lodi, "The feasibility pump," *Mathematical Programming*, vol. 104, no. 1, pp. 91–104, 2005.

[10] L. Bertacco, M. Fischetti, and A. Lodi, "A feasibility pump heuristic for general mixed-integer problems," *Discrete Optimization*, vol. 4, no. 1, pp. 63–76, 2007.

[11] S. Dey, A. Iroume, M. Molinaro, and D. Salvagnin, "Improving the randomization step in feasibility pump," *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 355–378, 2018.

[12] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt, "Penalty alternating direction methods for mixed-integer optimization: A new view on feasibility pumps," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1611–1636, 2017.

[13] S. Dey, A. Iroume, M. Molinaro, and D. Salvagnin, "Exploiting sparsity of MILPs by improving the randomization step in feasibility pump," *SIAM Journal on Optimization,(to appear)*, 2016.

[14] K. Huang and S. Mehrotra, "An empirical evaluation of walk-and-round heuristics for mixed integer linear programs," *Computational Optimization and Applications*, vol. 55, no. 3, pp. 545–570, 2013.

[15] N. Boland, A. Eberhard, F. Engineer, M. Fischetti, M. Savelsbergh, and A. Tsoukalas, "Boosting the feasibility pump," *Mathematical Programming Computation*, vol. 6, no. 3, pp. 255–279, 2014.

[16] F. I. Miertoiu and B. Dumitrescu, "Feasibility pump algorithm for sparse representation under laplacian noise," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[17] F. I. Miertoiu and B. Dumitrescu, "Feasibility pump algorithm for sparse representation under gaussian noise," *Algorithms*, vol. 13, no. 4, p. 88, 2020.

[18] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau, "Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance," *IEEE Trans. Signal Proc.*, vol. 64, no. 6, pp. 1405–1419, 2016.

[19] T. Achterberg and T. Berthold, "Improving the feasibility pump," *Discrete Optimization*, vol. 4, no. 1, pp. 77–86, 2007.

[20] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1." http://cvxr.com/cvx, Mar. 2014.

[21] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *27th Asilomar Conf. Signals Systems Computers*, vol. 1, pp. 40–44, Nov. 1993.

[22] A. Javaheri, H. Zayyani, M. A. Figueiredo, and F. Marvasti, "Robust sparse recovery in impulsive noise via continuous mixed norm," *IEEE Signal Proc. Letters*, vol. 25, no. 8, pp. 1146–1150, 2018.

[23] H. Zayyani, "Continuous mixed p-norm adaptive algorithm for system identification," *IEEE Signal Proc. Letters*, vol. 21, no. 9, pp. 1108–1110, 2014.

[24] L. Wang, M. D. Gordon, and J. Zhu, "Regularized least absolute deviations regression and an efficient algorithm for parameter tuning," in *6th Int. Conf. Data Mining*, pp. 690–700, IEEE, 2006.