

STABLE RECOVERY OF SPARSE VECTORS FROM RANDOM SINUSOIDAL FEATURE MAPS

Mohammadreza Soltani and Chinmay Hegde

Department of Electrical and Computer Engineering, Iowa State University

ABSTRACT

Random sinusoidal features are a popular approach for speeding up kernel-based inference in large datasets. Prior to the inference stage, the approach suggests performing dimensionality reduction by first multiplying each data vector by a random Gaussian matrix, and then computing an element-wise sinusoid. Theoretical analysis shows that collecting a sufficient number of such features can be reliably used for subsequent inference in kernel classification and regression.

In this work, we demonstrate that with a mild increase in the dimension of the embedding, it is also possible to reconstruct the data vector from such random sinusoidal features, provided that the underlying data is sparse enough. In particular, we propose a numerically stable algorithm for reconstructing the data vector given the nonlinear features, and analyze its sample complexity. Our algorithm can be extended to other types of structured inverse problems, such as demixing a pair of sparse (but incoherent) vectors. We support the efficacy of our approach via numerical experiments.

Index Terms— random features, sparsity, nonlinear recovery

1. INTRODUCTION

1.1. Setup

Several popular techniques in statistical regression and classification rely on the *kernel trick*, which enables approximating complicated nonlinear functions given a sufficiently large number of training samples. However, in large-scale problems where the number as well as dimension of the training data points is high, the computational costs of kernel-based techniques can be severe.

One approach to alleviate these computational costs is via the use of *random sinusoidal feature maps*, pioneered by [1]. The mechanism is conceptually simple. Suppose that the kernel of choice is the Gaussian (RBF) kernel. Prior to the inference stage, the approach suggests performing a *dimensionality reduction* of a given data vector by first multiplying by a random Gaussian matrix, followed by the application of a nonlinearity (specifically, a sinusoid or complex exponential). Mathematically, one can represent the above process as follows. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be any set in the data space, and consider a data vector $x \in \mathcal{X}$. Construct a matrix $B \in \mathbb{R}^{q \times n}$ whose entries are independently from a normal distribution. Then, modulo scaling factors, the random sinusoidal feature map, $y = \mathcal{A}(x)$, can be represented as the composition of the linear map $A \in \mathbb{R}^{m \times n}$ with a (complex) sinusoidal nonlinearity (see section 2 for more details):

$$z_j = (Ax)_j, \quad y_j = \exp(i z_j), \quad j = 1 \dots m. \quad (1.1)$$

One can alternately replace the complex exponential by a real-valued sinusoidal function with similar consequences. The authors

of [1] show that collecting a sufficient number of such features leads to statistically robust performance of kernel-based learning methods. Subsequent works [2, 3, 4, 5] have progressively extended this approach to more general types of kernels, faster methods of dimensionality reduction, as well as sparsity assumptions on the training data [6, 7].

Moving beyond inference, a natural question arises whether the original data vector x can at all be *reconstructed* from the nonlinear features y . Geometrically, we would like to understand if (and how) the random feature map can be efficiently inverted, given the observations y and knowledge of the dimensionality reduction operator B . This question is of both theoretical and practical interest; if the embedding is shown to be *reversible*, then it could have implications for the privacy-preserving properties of random feature maps.

Observe that (1.1) is analogous to the well-studied statistical notion of a generalized linear model (GLM) with a sinusoidal *link* (or transfer) function. Here, the data vector $x \in \mathbb{R}^n$ represents the model parameters. Reconstruction of high dimensional vectors in the GLM setting has been an intense focus of study in the statistical learning literature in recent years; applications of such recovery methods span scalable machine learning [8], 1-bit compressive sensing [9, 10], and imaging [11]. In signal processing applications, observation models with periodic link functions have been proposed for scalar quantization [12] as well as computational imaging [13].

1.2. Contributions

We consider a slightly more general version of the model (1.1) where the features y can be corrupted by noise. Moreover, we assume that the data vector x is s -sparse, i.e., it contains no more than s non-zeros. For this model, we propose an efficient, two-stage algorithm for reconstructing signals from a small number of random sinusoidal features (in particular, far smaller than the dimension of the data). To the best of our knowledge, our method is the first to propose an algorithm that is specialized to reconstructing sparse vectors from random sinusoidal feature maps.

To supplement our algorithm, we provide a theoretical sample complexity analysis. Our theory demonstrates that stable recovery from nonlinear sinusoidal observations can be done with *essentially* the same number of samples as standard sparse recovery with linear observations. The only additional cost is a logarithmic factor overhead in the dimension of the embedding that depends on n and the Euclidean norm of the original data vector. We also support the theory and algorithms via a range of numerical experiments.

1.3. Techniques

There are two main challenges that we will need to overcome. The first challenge is due to the ill-posed nature of the problem; in the high-dimensional regime, the total number of samples can be far

This work is supported in part by the National Science Foundation under the grants CCF-1566281 and IIP-1632116.

less than the native dimension of the data. The second (and more compelling) challenge arises due to the fact that the sinusoidal transfer function is highly non-invertible. The typical way to deal with this is to assume some upper bound on the magnitude of the entries of the linear projection (Ax). However, we observe that unless this upper bound is relatively small, we can only estimate z_j up to some unknown integer multiple of 2π . If the projected dimension of the linear embedding, q , is large, then the number of possible combinations becomes exponential in q .

We alleviate these challenges using a simple idea. The key is to *decouple* the problem of inverting the sinusoidal nonlinearity from the sparse recovery step by replacing the linear map A with a different (carefully designed) map that still possesses all the properties required for reliable kernel-based inference. Particularly, our new linear map A is the product of two matrices $D \cdot B$; the way to construct D and B is described in detail below. This technique has been inspired by some recent approaches for the related problem of *phase retrieval* [14, 15]. Using this decoupling technique, we can separately solve the inversion of the nonlinearity and the actual recovery of the sparse high-dimensional data in two stages.

For the first stage, we *individually* estimate the linear projection $z = Bx$ using classical signal processing techniques for line spectral estimation. In particular, we leverage the method of matched filtering from random samples [16]. In the absence of noise, any spectral estimation technique (such as MUSIC, root-MUSIC, and ESPRIT) can also be used; however, our Monte Carlo simulations show below that the randomized method of [16] et al. is considerably more robust compared to these more classical methods. For the second stage, we can use any sparse recovery method of our choice. In our experiments, we have used the CoSaMP algorithm proposed in [17].

While conceptually simple, the generic nature of our algorithm is advantageous in many ways. From a theoretical perspective, mirroring the approach of [14], we can combine existing results for robust line spectral estimation with robust sparse recovery and obtain guarantees on the sample complexity and robustness of our algorithm. Moreover, since the nonlinear inversion and the sparse recovery steps are decoupled, one could conceivably extend this decoupling approach to a variety of other signal models and recovery algorithms (such as structured-sparsity [18, 19], low-rank matrix models [20], and demixing a pair of sparse incoherent vectors [21, 22, 23]).

1.4. Comparison with prior work

Our approach relies upon ideas from several related problems in machine learning and signal processing. The technique of using random projections in conjunction with sinusoidal nonlinearities has emerged as a popular alternative for kernel-based inference [1, 2, 3, 4, 5]. However, these works do not explicitly consider recovering the original data from the nonlinear random projections.

In parallel, there has been considerable progress for various problems where *reconstruction* of data vectors having some form of low-dimensional structure is of chief interest [24, 25, 26, 27, 28]. Particularly, there has been considerable recent interest in reconstructing from nonlinear measurements (features). However, the majority of the methods deal with the case where the nonlinearity is either monotonic [8] or satisfies restricted regularity conditions [28, 27].

None of the proposed methods in the literature can be directly applied for the specific problem of stable sparse reconstruction from random sinusoidal features, with one exception: the theoretical framework developed in [25] can indeed be adapted to our problem. However, the recovery algorithm only yields an estimate up to a scalar ambiguity, which can be problematic in applications. Moreover, even

if scalar ambiguities can be tolerated, we show in our experimental results below that the sample complexity of this method is far too high in practice. In contrast, our proposed method yields stable and accurate results with only a mild factor increase in sample complexity when compared to standard sparse recovery.

2. MATHEMATICAL MODEL AND ALGORITHM

In this section, we present our algorithm for solving problem (1.1) and provide the theory for sample complexity of the proposed algorithm. We first establish some notation. Let the j^{th} entry of the signal $x \in \mathbb{R}^n$ be denoted as x_j . For any $j \in [q]$, $x(j : (k-1)q + j)$ denotes the sub-vector in \mathbb{R}^k formed by the entries of x starting at index $j + qr$, where $r = 0, 1, \dots, k-1$. Similarly, $A((j : q : (k-1)q, l))$ represents the sub-vector constructed by picking the l^{th} column of any matrix A and selecting the entries of this column with the same procedure as mentioned.

As the key component in our approach, instead of using a random Gaussian linear projection as proposed in [1], we construct a linear operator A that can be *factorized* as $A = DB$, where $D \in \mathbb{R}^{m \times q}$, and $B \in \mathbb{R}^{q \times n}$. We assume that m is a multiple of q , and that D is a concatenation of k diagonal matrices of $q \times q$ such that the diagonal entries in the blocks of D are i.i.d. random variables, drawn from a distribution that we specify later. The choice of B depends on the model assumed on the data vector x ; if the data is s -sparse, then B can be any matrix that supports stable sparse recovery (more precisely, B satisfies the null-space property [29]). Overall, our low-dimensional feature map can be written as:

$$y = \exp(iDBx) + e = \exp \left(i \begin{bmatrix} D^1 \\ D^2 \\ \vdots \\ D^k \end{bmatrix} Bx \right) + e, \quad (2.1)$$

where D^i are diagonal matrices and $e \in \mathbb{R}^m$ denotes additive noise such that $e \sim \mathcal{N}(0, \sigma^2 I)$. The goal is to stably recover x from the embedding y .

By the block diagonal structure of the outer projection D , we can reduce the overall reconstruction problem to first obtaining a good enough estimate of each entry of Bx . We show below that this can be achieved using line spectral estimation methods. The output of the first stage is used as the input of the second stage, which involves estimating x from a (possibly noisy) estimate of Bx .

2.1. Line spectral estimation

Consider the observation model (2.1) and let $z = Bx \in \mathbb{R}^q$. Suppose we know *a priori* that the entries of z belong to some bounded set $\Omega \in \mathbb{R}$. Fix $l \in [q]$, and let $t = D(l : q : (k-1)q + l)$, $u = y(l : q : (k-1)q + l)$, $h = e(l : q : (k-1)q + l)$ which are vectors in \mathbb{R}^k . We observe that:

$$u = \exp(iz_l t) + h.$$

In other words, u can be interpreted as a collection of time samples of a (single-tone, potentially off-grid) complex-valued signal with frequency $z_l \in \Omega$, measured at time locations t . Therefore, we can independently estimate z_l for $l = 1, \dots, q$ by solving a least-squares problem [16]:

$$\hat{z}_l = \arg \min_{v \in \Omega} \|u - \exp(ivt)\|_2^2 = \arg \max_{v \in \Omega} |\langle u, \psi_v \rangle|, \quad (2.2)$$

for all $l = 1, \dots, q$, where $\psi_v \in \mathbb{R}^k$ denotes a *template vector* given by $\psi_v = \exp(jtv)$ for any $v \in \Omega$. In essence, the solution of the least-squares problem can be interpreted as a *matched filter*. Numerically, the optimization problem in (2.2) can be solved using a grid search over the set Ω , and the resolution of this grid search controls the running time of the algorithm; for fine enough resolution, the estimation of z_l is more accurate albeit with increased running time. This issue is also discussed in [16] and more sophisticated spectral estimation techniques have appeared in [30, 31, 32]. After obtaining all the estimates \hat{z}_l 's, we stack them in a vector \hat{z} .

2.2. Sparse recovery

We now propose to recover the high-dimensional signal $x \in \mathbb{R}^m$ in problem (2.1) using our estimated $\hat{z} \in \mathbb{R}^q$. According to our model, we also have assumed that the matrix B supports stable sparse recovery, and the underlying signal x is s -sparse. Hence, we can use any generic sparse recovery algorithm of our choice to obtain an estimate of x . In our simulations below, we use the CoSaMP algorithm [17] due to its speed and ease of parameter tuning. Again, we stress that this stage depends on a model assumption on x , and other recovery algorithms (such as structured-sparse recovery [18], low-rank matrix recovery [20], or demixing a pair of sparse (but incoherent) vectors [21]) can equally well be used. Our overall algorithm, *Matched Filtering+Sparse recovery* (MF-Sparse) is described in pseudocode form in Algorithm 1. We now provide a theoretical analysis of our proposed algorithm. Our result follows from a concatenation of the results of [16] and [17].

Theorem 2.1 (Sample complexity of MF-Sparse). *Assume that the nonzero entries of D are i.i.d. samples from a uniform distribution $[-T, T]$, and the entries of B are i.i.d. samples from $\mathcal{N}(0, 1/q)$. Also, assume $\|x\|_2 \leq R$ for some constant $R > 0$. Set $m = kq$ where $k = c_1 \log(\frac{Rq}{\epsilon}) (1 + \sigma^2)$ for some $\epsilon > 0$ and $q = c_2 (s \log \frac{n}{s})$. Set $\omega = c_3 R$ and $\Omega = [-\omega, \omega]$. Then, MF-Sparse returns an estimate \hat{x} , such that*

$$\|x - \hat{x}\|_2 \leq C\epsilon,$$

with probability exceeding $1 - \delta$. Here, c_1, c_2, c_3, C are constants.

Proof. Set $\epsilon' > 0$. Based on Corollary 8 of [16], by setting $|T| = \mathcal{O}(\frac{1}{\epsilon'})$, we need $k = \mathcal{O}\left((1 + \sigma^2) \log\left(\frac{|\Omega|}{\epsilon'} \frac{1}{\delta}\right)\right)$ to obtain $|\hat{z}_l - z_l| \leq \epsilon'$ with probability at least $1 - \delta$ for each $l = 1, \dots, q$. Here, $|\Omega|$ denotes the radius of the feasible range for z , and appears in the maximization problem (2.2). Observe that $|\Omega| = \mathcal{O}(\|Bx\|_\infty)$. If the entries of matrix B are chosen as stipulated in the theorem, then $\|Bx\|_\infty \leq \|Bx\|_2 \lesssim \mathcal{O}(R)$ which justifies the choice of $|\Omega|$ in the theorem. Thus, $k = \mathcal{O}\left((1 + \sigma^2) \log\left(\frac{R}{\epsilon'} \frac{1}{\delta}\right)\right)$. By a union bound, it follows that with probability at least $1 - q\delta$, we have $\|\hat{z} - z\|_\infty \leq \epsilon'$. Now, we can write $\hat{z} = z + e' = Bx + e'$ where $\|e'\|_\infty \leq \epsilon'$. Since we have used CoSaMP in the sparse recovery stage, if the choice of $q = \mathcal{O}(s \log \frac{n}{s})$ enables us to obtain $\|\hat{x} - x\|_2 \leq c\|e'\|_2 \leq c\sqrt{q}\|e'\|_\infty \leq c\sqrt{q}\epsilon'$ [17]. (In fact, a more general guarantee can be derived even for the case when x is not exactly s -sparse.) Now, let $\epsilon' = \mathcal{O}\left(\frac{\epsilon}{\sqrt{q}}\right)$ to obtain the final bound on the estimation error $\|\hat{x} - x\|_2 \leq \mathcal{O}(\epsilon)$ with $k = \mathcal{O}\left((1 + \sigma^2) \log\left(\frac{Rq}{\epsilon} \frac{1}{\delta}\right)\right)$. \square

In model (1.1), the features y_j are modeled in terms of complex exponentials. With only a slight modification in the line spectral estimation stage, we can recover x from real-valued random sine features. More precisely, these random features are given by:

$$y = \sin(DBx) + e \quad (2.3)$$

Algorithm 1 MF-SPARSE

Inputs: y, D, B, Ω, s

Output: \hat{x}

Stage 1: Tone estimation:

for $l = 1 : q$ **do**

$t \leftarrow D(l : q : (k-1)q + l, l)$

$u \leftarrow y(l : q : (k-1)q + l)$

$\hat{z}_l = \arg \max_{\omega \in \Omega} |\langle y, \psi_\omega \rangle|$

end for

$\hat{z} \leftarrow [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_q]^T$

Stage 2: Sparse recovery

$\hat{x} \leftarrow \text{CoSaMP}(\hat{z}, B, s)$

where e denotes the additive noise as defined before. If we follow an analogous approach for estimating x , then in lieu of the least squares estimator (2.2), we have the following estimator:

$$\hat{z}_l = \arg \min_{v \in \Omega} \|u - \sin(vt)\|_2^2 \quad (2.4)$$

$$= \arg \max_{v \in \Omega} (2|\langle u, \psi_v \rangle| - \|\psi_v\|_2^2), \quad (2.5)$$

for $l = 1, \dots, q$ and u as defined above. Also, $\psi_v = \sin(tv)$ for any $v \in \Omega$. We only provide some numerical experiments for this estimator and leave a detailed theoretical analysis for future research.

3. EXPERIMENTAL RESULTS

We compare our algorithm with existing algorithms in various scenarios. In the first experiment, we assume that our random features are computed using a real sine link function. In other words, the features, $\{y_j\}$ is given by (2.3) for all $j = 1, \dots, q$.

In Fig. 1(a), we compare the probability of recovery of MF-Sparse with *Gradient Hard Thresholding* (GHT), a projected-gradient descent type algorithm whose variants have proposed in [28, 33, 34]. To recover x , GHT tries to minimize a specific loss function (typically, the squared loss) between the observed random feature vector, y , and $\sin(DBx)$ by iteratively updating the current estimate of x based on a gradient update rule, and then projecting it into the set of s -sparse vectors via hard thresholding.

The setup for the experiment illustrated in Fig. 1(a) is as follows. First, we generate a synthetic signal of length $n = 2^{14}$ with sparsity $s = 100$ such that the support is random and the values of the signal in the support are drawn from a standard normal distribution. Then, the ℓ_2 -norm of x is adjusted via a global scaling to coincide with three different values; $\|x\|_2 = 1$, $\|x\|_2 = 15$, and $\|x\|_2 = 30$. We generate a matrix $B \in \mathbb{R}^{q \times n}$ where $q = 700$ and $n = 2^{14}$ such that the entries of it are i.i.d random variables with distribution $\mathcal{N}(0, \frac{1}{\sqrt{q}})$. All nonzero entries of D, d_l^r for $l = 1, \dots, q$ and $r = 1, \dots, k$ are assumed to be standard normal random variables for $k = 1, \dots, 8$. Next, we generate $y \in \mathbb{R}^m$ as $y = \sin(DBx)$ where $m = kq$. (There is no noise considered in this experiment.) By running MF-Sparse and GHT, we obtain the estimate of x , denoted by \hat{x} , and calculate the (normalized) estimation error defined as $\frac{\|\hat{x} - x\|_2}{\|x\|_2}$. We repeat this process for 60 Monte Carlo trials, and define the empirical probability of successful recovery as the fraction of simulations in which the relative error is less than 0.05.

As we can see in Fig. 1(a), MF-Sparse can successfully recover x even with $k = 4$ number of blocks with probability close to 1 when the norm of x is small. In this regime, i.e., $\|x\|_2$ being small, both GHT and MF-Sparse display similar performance. However, GHT

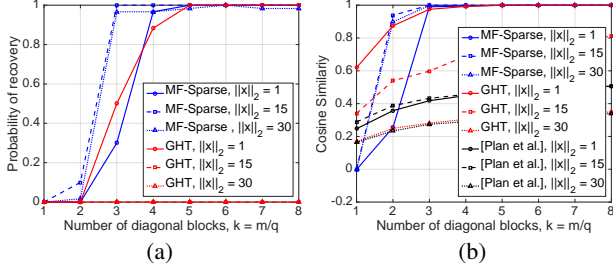


Fig. 1: Comparison of the proposed algorithm with other algorithms. Parameters: $n = 2^{14}$, $q = 700$. (a) Probability of recovery in terms of normalized error. (b) Cosine similarity between recovered and true signal.

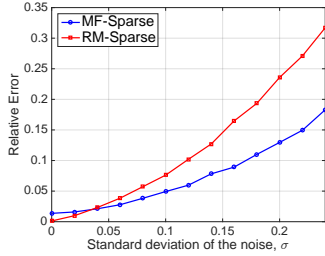


Fig. 2: Comparison of matched filtering versus rootMUSIC in the first stage. Parameters: $n = 2^{14}$, $q = 800$, $k = 6$.

shows extremely poor performance when $\|x\|_2$ has moderate or large value. The reason is that when the norm of x is small, the entries of Bx are also small (i.e., the entries of DBx are close to the origin with high probability), and the sinusoidal nonlinearity can be assumed to be almost linear. Consequently, GHT can recover x in this regime. However, this assumption breaks down for larger values of $\|x\|_2$.

In Fig. 1(b), we repeat a similar experiment, but measure performance with a different recovery criterion. In this scenario, we measure the *cosine similarity* of the estimated vector \hat{x} with x , defined as $\frac{\hat{x}^T x}{\|\hat{x}\|_2 \|x\|_2}$. In addition to GHT, we also compare the performance of MF-Sparse with the single-step thresholding approach proposed by [25]. The approach of [25] only recovers the vector modulo an (unknown) scaling factor, and does not need to possess knowledge of the nonlinearity. As illustrated in Fig. 1(b), the approach of [25] has worse performance compared to two other methods. Also, GHT shows good performance only when $\|x\|_2 = 1$, as expected. In contrast, MF-Sparse shows the best performance.

Next, we consider the experiment illustrated in Fig. 2. In this experiment, we assume a complex sinusoid as the nonlinearity generating the feature map. In this scenario, we fix $k = 6$ (number of diagonal blocks in matrix D), $q = 800$, and add i.i.d. Gaussian noise to the embeddings with increasing variance. Our goal is to compare the matched filtering part of MF-Sparse with classical spectral estimation techniques. Indeed, a natural question is whether we can use other spectral estimation approaches instead of the matched filter, and whether there is any advantage in choosing the elements of D randomly. We attempt to illustrate the benefits of randomness through the experiment in Fig. 2. Here, we compare the relative error of MF-Sparse with an analogous algorithm that we call *RM-Sparse*, which uses the RootMUSIC spectral estimation technique [36]. For RM-Sparse, in the line spectral estimation stage, we replace random diagonal blocks in matrix D with deterministic diagonal ones, and letting the diagonal entries of the r^{th} block being proportional to r . In other words, the time samples are chosen to lie on a uniform

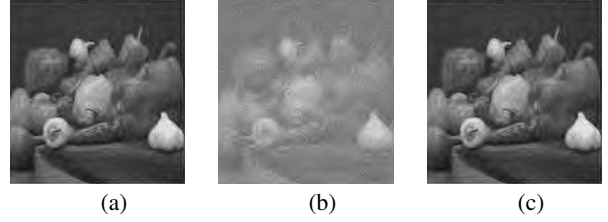


Fig. 3: Successful reconstruction on a real 2-D image from random sinusoidal features. (a) Test image. (b) Reconstruction quality with $k = 2$ diagonal blocks. (c) Reconstruction with $k = 3$.

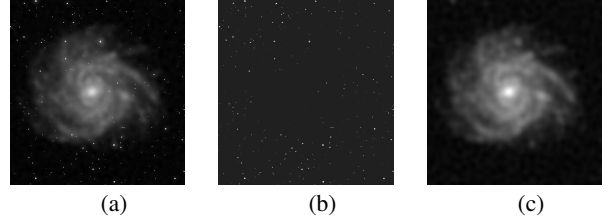


Fig. 4: Successful demixing on a real 2-dimensional image from random sinusoidal features. Parameters: $n = 512 \times 512$, $s = 1000$, $m = k \times q = 48000$, $g(x) = \sin(x)$. Image credits: [35].

grid, as RootMUSIC expects. As we see in Fig. 2, while RM-Sparse does recover the original x for very low values of noise variance, MF-Sparse outperforms RM-Sparse when the noise level increases; this verifies the robustness of MF-Sparse to noise in the observations.

In addition, we evaluate our proposed algorithm for a real 2D image. We begin with a 512×512 test image. First, we obtain its 2D Haar wavelet decomposition and sparsify it by retaining the $s = 2000$ largest coefficients. Then, we synthesize the test image based on these largest coefficients, and multiply it by a subsampled Fourier matrix with $q = 16000$ multiplied with a diagonal matrix with random ± 1 entries [37]. Further, we multiply this result with a block diagonal matrix D with number of blocks $k = 2, 3$. Now, we apply the $\sin(\cdot)$ function element-wise to the resulting vector. Figure 3 shows the recovered image using MF-Sparse. As is visually evident, with $k = 2$, the quality of the reconstructed image is poor, but $k = 3$ already yields a solution close to the true image.

Finally, we present another experiment on a real 2D image. In this case, we assume that our signal x is the superposition of two constituent signals, i.e., $x = \Phi w + \Psi z$. Hence, the random feature map is given by $y = \sin(DB(\Phi w + \Psi z))$. In this setting, as illustrated in Fig. 4(a), the signal x is the mixture of galaxy and star images, where the constituent coefficient vectors w and z are s -sparse signals in the DCT basis (Φ) and the canonical basis (Ψ). In this experiment, we fix $s = 1000$ for both w and z . The matrices B and D are generated the same way as in the experiment expressed in Fig. 3 with $q = 16000$, $n = 2^{18}$, and $k = 3$. As we can see in Figs. 4(b) and 4(c), running a variant of MF-Sparse along with the DHT algorithm [21] for the second stage can successfully separate the galaxy from the stars, verifying the applicability of MF-Sparse for more generic structured inverse problems.

4. REFERENCES

- [1] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2007, pp. 1177–1184.
- [2] A. Vedaldi and A. Zisserman, "Efficient additive kernels via

- explicit feature maps,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 3, pp. 480–492, 2012.
- [3] Q. Shi, J. Petterson, G. Dror, J. Langford, A. Smola, and S. Vishwanathan, “Hash kernels for structured data,” *J. Machine Learning Research*, vol. 10, no. Nov, pp. 2615–2637, 2009.
 - [4] Q. Le, T. Sarlos, and A. Smola, “Fastfood-approximating kernel expansions in loglinear time,” in *Proc. Int. Conf. Machine Learning*, 2013.
 - [5] D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, and B. Schölkopf, “Randomized nonlinear component analysis,” in *Proc. Int. Conf. Machine Learning*, 2014.
 - [6] J. Chang, A. Sankaranarayanan, and B. V. K. Vijayakumar, “Random features for sparse signal classification,” in *IEEE Conf. Comp. Vision and Pattern Recog.*, 2016.
 - [7] M. Hosseini, A. Hajisami, and D. Pompili, “Real-time epileptic seizure detection from eeg signals via random subspace ensemble learning,” in *IEEE Int. Conf. Autonomic Computing (ICAC)*, 2016, pp. 209–218.
 - [8] S. Kakade, V. Kanade, O. Shamir, and A. Kalai, “Efficient learning of generalized linear and single index models with isotonic regression,” in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2011, pp. 927–935.
 - [9] P. Boufounos and R. Baraniuk, “1-bit compressive sensing,” in *Int. Conf. Info. Sciences and Systems (CISS)*. IEEE, 2008, pp. 16–21.
 - [10] L. Jacques, J. Laska, P. Boufounos, and R. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Trans. Inform. Theory*, vol. 59, no. 4, pp. 2082–2102, 2013.
 - [11] E. Candes, X. Li, and M. Soltanolkotabi, “Phase retrieval via wirtinger flow: Theory and algorithms,” *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.
 - [12] P. Boufounos, “Universal rate-efficient scalar quantization,” *IEEE Trans. Inform. Theory*, vol. 58, no. 3, pp. 1861–1872, 2012.
 - [13] H. Zhao, B. Shi, C. Fernandez-Cull, S.-K. Yeung, and R. Raskar, “Unbounded high dynamic range photography using a modulo camera,” in *Proc. Int. Conf. Comp. Photography*, 2015.
 - [14] M. Iwen, A. Viswanathan, and Y. Wang, “Robust sparse phase retrieval made easy,” *Appl. Comput. Harmon. Anal.*, 2015.
 - [15] S. Bahmani and J. Romberg, “Efficient compressive phase retrieval with constrained sensing vectors,” in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2015, pp. 523–531.
 - [16] A. Eftekhari, J. Romberg, and M. Wakin, “Matched filtering from limited frequency samples,” *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3475–3496, 2013.
 - [17] D. Needell and J. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.
 - [18] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, “Model-based compressive sensing,” *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.
 - [19] C. Hegde, P. Indyk, and L. Schmidt, “Approximation algorithms for model-based compressive sensing,” *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 5129–5147, 2015.
 - [20] B. Recht, M. Fazel, and P. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.
 - [21] M. Soltani and C. Hegde, “Fast algorithms for demixing sparse signals from nonlinear observations,” *arXiv preprint arXiv:1608.01234*, 2016.
 - [22] S. Minaee and Y. Wang, “Screen content image segmentation using robust regression and sparse decomposition,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 573–584, 2016.
 - [23] S. Minaee, A. Abdolrashidi, and Y. Wang, “Screen content image segmentation using sparse-smooth decomposition,” in *Proc. Asilomar Conf. Sig. Sys. Comp.*, 2015, pp. 1202–1206.
 - [24] A. Beck and Y. Eldar, “Sparsity constrained nonlinear optimization: Optimality conditions and algorithms,” *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1480–1509, 2013.
 - [25] Y. Plan, R. Vershynin, and E. Yudovina, “High-dimensional estimation with geometric constraints,” *arXiv preprint arXiv:1404.3749*, 2014.
 - [26] X. Yi, Z. Wang, C. Caramanis, and H. Liu, “Optimal linear estimation under unknown nonlinear transform,” in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2015, pp. 1549–1557.
 - [27] Z. Yang, Z. Wang, H. Liu, Y. Eldar, and T. Zhang, “Sparse nonlinear regression: Parameter estimation and asymptotic inference,” *arXiv preprint arXiv:1511.04514*, 2015.
 - [28] S. Bahmani, P. Boufounos, and B. Raj, “Greedy sparsity-constrained optimization,” *JMLR*, 2013.
 - [29] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*, Springer, 2013.
 - [30] M. Mishali and Y. Eldar, “Wideband spectrum sensing at subnyquist rates [applications corner],” *IEEE Sig. Proc. Mag.*, vol. 28, no. 4, pp. 102–135, 2011.
 - [31] G. Tang, B. Bhaskar, P. Shah, and B. Recht, “Compressed sensing off the grid,” *IEEE Trans. Inform. Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.
 - [32] Y. Chen and Y. Chi, “Robust spectral compressed sensing via structured matrix completion,” *IEEE Trans. Inform. Theory*, vol. 60, no. 10, 2014.
 - [33] X. Yuan, P. Li, and T. Zhang, “Gradient hard thresholding pursuit for sparsity-constrained optimization,” in *Proc. Int. Conf. Machine Learning*, 2014, pp. 127–135.
 - [34] P. Jain, A. Tewari, and P. Kar, “On iterative hard thresholding methods for high-dimensional m-estimation,” in *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2014, pp. 685–693.
 - [35] M. McCoy, V. Cevher, Q. Dinh, A. Asaei, and L. Baldassarre, “Convexity in source separation: Models, geometry, and algorithms,” *IEEE Sig. Proc. Mag.*, vol. 31, no. 3, pp. 87–95, 2014.
 - [36] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. Antennas and Prop.*, vol. 34, no. 3, pp. 276–280, 1986.
 - [37] F. Krahmer and R. Ward, “New and improved johnson-lindenstrauss embeddings via the restricted isometry property,” *SIAM J. Math. Anal.*, vol. 43, no. 3, pp. 1269–1281, 2011.