

# K-SVD based Periodicity Dictionary Learning

Pranav Kulkarni and P. P. Vaidyanathan

Department of Electrical Engineering

California Institute of Technology

pkulkarn@caltech.edu, ppvnath@systems.caltech.edu

**Abstract**—It has recently been shown that periodicity in discrete-time data can be analyzed using Ramanujan sums and associated dictionaries. This paper explores the role of dictionary learning methods in the context of period estimation and periodic signal representation using dictionaries. It is shown that a well-known dictionary learning algorithm, namely K-SVD, is able to learn Ramanujan and Farey periodicity dictionaries from the noisy, sparse coefficient data generated from them without imposing any periodicity structure in the learning stage. This similarity between the learned dictionary and the underlying original periodicity dictionary reaffirms the power of the K-SVD in predicting the right dictionary from data without explicit application-specific constraints. The paper also examines how the choice of different parameter values affect the similarity of the learned dictionary to the underlying dictionary. Two versions of K-SVD along with different initializations are analyzed for their effect on representation and denoising error for the data.

**Index Terms**—Nested periodic dictionaries, dictionary learning, K-SVD, period estimation, denoising.

## I. INTRODUCTION

It is common in many signal processing applications to use overcomplete dictionaries to represent data [1]–[6]. A good review of different dictionary types can be found in [1]. One way to construct representation dictionaries is to design them based on the a-priori knowledge or mathematical model regarding the signals that are to be represented. Such dictionaries are called *analytical dictionaries*. These dictionaries often have the advantage of fast implicit implementation. Especially if the  $m \times n$  dictionary  $D$  is designed to be a tight frame, or equivalently  $DD^T x = \frac{n}{m} x$  for all  $x$ , then  $D^T$  is a possible analysis operator that produces analysis coefficients. Some examples of analytical dictionaries are curvelets, countourlets, and bandlets.

Another way to develop dictionaries is from a set of realizations of the data. The advantage of such trained dictionaries that are directly learned from data is that they adapt to the non-idealities of the data well instead of relying on inaccurate modeling of these non-idealities. There are many dictionary learning algorithms such as K-SVD [5], Method of Optimal Directions [7], Union of Orthonormal Basis [8], and so on. Overcomplete dictionaries are widely used along with sparse vector recovery formulation for many applications such as denoising [6], the direction of arrival estimation [4], and data compression [3] to name a few. In addition to analytical and learned dictionaries, there is also a class of parametric dictionaries [9]. Atoms of these dictionaries are structured but have a few free parameters that can be optimized by the training algorithm according to the data.

This work was supported in parts by the ONR grant N00014-18-1-2390, the NSF grant CCF-1712633, and the California Institute of Technology.

It has been shown in recent years that Ramanujan sums are useful in identifying periodicity structures in signals. The dictionaries used for this purpose are analytical dictionaries such as Ramanujan and Farey dictionaries [10], [11]. A comprehensive review of Ramanujan sums for detecting periods in a discrete-time signal can be found in [10], [12], [13]. In this paper, we propose to use the dictionary learning methods in the context of period estimation and periodic signal representation using dictionaries. We investigate whether K-SVD can learn the known dictionaries for period estimation from sparse coefficient periodic data generated from Ramanujan and Farey dictionaries. We also experimentally analyze the dependence of the similarity of the learned dictionary and the true underlying dictionary with different parameters.

### A. Preliminaries and Notation

A discrete-time periodic signal  $x(n)$  is periodic with period  $P$  if  $P$  is the smallest possible integer such that

$$x(n + P) = x(n) \quad \forall n \in \mathbb{Z} \quad (1)$$

For all integers  $q > 0$ , the  $q$ -th Ramanujan sum is defined as [14]

$$c_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^q e^{j2\pi kn/q} \quad (2)$$

Here the notation  $(k, q)$  denotes the gcd of the integers  $k$  and  $q$ , so that  $(k, q) = 1$  means that  $k$  and  $q$  are coprime. It is well known [12] that  $c_q(n)$  is integer valued and periodic with period  $q$ . The  $q$ -th Ramanujan sum lies in the  $q$ -th Ramanujan subspace defined as [12]

$$S_q = \text{span}\{e^{j2\pi kn/q}, (k, q) = 1\} \quad (3)$$

Note that  $S_q$  is a  $\phi(q)$  dimensional subspace. It contains all the signals of period  $q$  that have possibly non-zero DFT value only at the frequency indices  $k$  that are coprime to  $q$ .  $S_q$  also admits an integer basis in terms of the  $q$ -th Ramanujan sum and its first  $\phi(q)$  circularly shifted versions [12]

$$S_q = \text{span}\{c_q(n - l), 0 \leq l \leq \phi(q) - 1\} \quad (4)$$

Here,  $\phi(q)$  is the Euler's totient function that denotes the number of integers  $k$  in  $1 \leq k \leq q$  satisfying  $(k, q) = 1$ . We define  $\Phi(q) = \sum_{i=1}^q \phi(i)$  as the sum of Euler's totient function from 1 to  $q$ .

### B. Organization of the Paper

In Sec. II we review the two components that are the focus of this paper, namely, the known dictionaries for period estimation and a popular dictionary learning algorithm, the K-SVD. In Sec. III we formalize the goals of this paper and describe the experimental setting. Sec. IV presents experimental results and discussions. Sec. V concludes the paper.

## II. REVIEW OF KNOWN PERIODICITY DICTIONARIES AND THE DICTIONARY LEARNING ALGORITHM K-SVD

### A. Nested Periodic Dictionaries for Period Estimation

Any length  $N$  periodic signal  $\mathbf{x}$  with period  $P \leq N$  can be represented as [12]  $\mathbf{x} = \mathbf{F}_N \mathbf{b}$ , where

$$\mathbf{F}_N = [\mathbf{G}_{q_1} \ \mathbf{G}_{q_2} \ \dots \ \mathbf{G}_{q_k}] \quad (5)$$

Here,  $q_1, q_2, \dots, q_k$  are the divisors of signal length  $N$ . The matrices  $\mathbf{G}_{q_i}$  have the  $\phi(q_i)$  circularly shifted versions of the  $q_i$ -th Ramanujan sums as the columns and hence span the corresponding periodic subspaces  $S_{q_i}$ . The columns are periodically repeated to make the number of rows equal to  $N$ . This representation is called the Ramanujan Periodic Transform (RPT). The component or hidden periods can be found by identifying the set of divisors  $\{q_i\}$  that have corresponding non-zero entries in the coefficient vector  $\mathbf{b}$ .

Although the RPT is shown to have some applications, it is useful only when the component periods of the signal are the divisors of the signal length  $N$ . Generalizing the transform matrix for RPT, a framework of nested periodic matrices and dictionaries was proposed in [11]. Namely, the above matrix  $\mathbf{F}_N$  was replaced with full rank matrix  $\mathbf{A}$  defined as

$$\mathbf{A} = [\mathbf{H}_{q_1} \ \mathbf{H}_{q_2} \ \dots \ \mathbf{H}_{q_k}] \quad (6)$$

where  $\mathbf{H}_{q_i}$  can be any  $N \times \phi(q_i)$  matrix containing columns of period  $q_i$ . Examples of nested periodic matrices are natural basis matrices, DFT matrices, and Ramanujan matrices. In order to identify periods that may not be divisors of signal length, the use of overcomplete Farey dictionaries was proposed in [15] and was extended to other periodicity matrices in [11]. Here the matrix  $\mathbf{A}$  in the above formulation is replaced with a dictionary:

$$\mathbf{D} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_{P_{max}}] \quad (7)$$

Here each  $\mathbf{H}_i$  contains  $\phi(i)$  columns of period  $i$ , hence all the periods from 1 to  $P_{max}$  are represented in this dictionary. The last periods of  $\mathbf{H}_i$  are truncated if needed in order to make their length equal to the signal length  $N$ . The period estimation problem is formulated as a sparse vector recovery problem in this setting. Examples of nested periodic dictionaries include Farey Dictionary which has columns from multiple DFT matrices, Ramanujan dictionary which has shifted Ramanujan sums as columns, and natural periodic dictionary which has columns from different identity matrices.

### B. Learning Dictionary via K-SVD

Many researchers over the last two decades have developed methods to learn dictionaries from the given data samples. The objective is to represent the data as sparse linear combinations of the columns of a dictionary. In this paper, we use K-SVD [5], which is one such popular dictionary learning algorithm. K-SVD is shown to have numerous applications including denoising [6], compression [5], and face recognition [16]. It is a generalization of the  $k$ -means clustering algorithm. The objective function of K-SVD is

$$\min_{\mathbf{D}, \mathbf{\Gamma}} \|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{subject to } \|\gamma_i\|_0 \leq T_0 \ \forall i. \quad (8)$$

Here,  $\mathbf{Y}$  is the training data matrix,  $\mathbf{D}$  is a dictionary, and  $\mathbf{\Gamma}$  is the coefficient matrix of the data with respect to the dictionary  $\mathbf{D}$ , and  $\gamma_i$  denotes the  $i$ -th column of  $\mathbf{\Gamma}$ . In this

sparsity constrained formulation, each coefficient vector  $\gamma_i$  is constrained to have at most  $T_0$  non-zero entries.

The K-SVD algorithm alternates between two steps. In the first stage of sparse coding, the dictionary  $\mathbf{D}$  is kept fixed and the coefficients  $\gamma_i$  are updated with the solution of  $l_0$  norm constrained error minimization problem

$$\hat{\gamma}_i = \min_{\gamma_i} \|\mathbf{Y}_i - \mathbf{D}\gamma_i\|_2^2 \quad \text{subject to } \|\gamma_i\|_0 \leq T_0 \quad (9)$$

As the  $l_0$  norm is NP-hard to deal with, approximate sparse coding algorithms like Orthogonal Matching Pursuit (OMP) [17] or FOCUSS [18] can be used at this stage. The optimal solution is recovered with a high probability by OMP if the sparsity  $T_0$  is low [19].

In the second stage, columns of the dictionary  $\mathbf{D}$  are updated successively. To update the  $i$ -th column, the representation error matrix without the  $i$ -th column is found out

$$\mathbf{E}_i = \mathbf{Y} - \sum_{j \neq i} \mathbf{d}_j \gamma_j^j \quad (10)$$

Here,  $\gamma_j^j$  denotes the  $j$ -th row of  $\mathbf{\Gamma}$ . To preserve sparsity constraint, a reduced error matrix  $\mathbf{E}_i^R$  is formed by keeping those columns of  $\mathbf{E}_i$  with indices from the set  $\mathcal{H}_i = \{j \mid \mathbf{\Gamma}_{ij} \neq 0\}$ . These indices correspond to examples that use the  $i$ -th column in their current representation. With this, the problem reduces to finding a rank one approximation of the restricted error matrix  $\mathbf{E}_i^R$ , which can be conveniently found from the SVD:  $\mathbf{E}_i^R = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^H$ . The superscript  $R$  is used to denote matrices or vectors restricted according to the set  $\mathcal{H}_i$ , whereas superscript  $H$  denotes the conjugate transpose of the matrix. The dictionary column is updated by the first left singular vector  $\mathbf{d}_i = \mathbf{u}_1$ , and the corresponding coefficient row is updated with the first right singular vector scaled with the largest singular value  $\gamma^i = \sigma_1 \mathbf{v}_1^H$ .

The K-SVD can also be formulated as an error constrained objective minimizing the sparsity.

$$\min_{\mathbf{D}, \mathbf{\Gamma}} \|\gamma_i\|_0 \quad \text{subject to } \|\mathbf{y}_i - \mathbf{D}\gamma_i\|_2 \leq \epsilon \ \forall i \quad (11)$$

Fast implementation of K-SVD has been developed in [20] using batch-OMP. The K-SVD can also be used for complex dictionary learning as in [21].

K-SVD is susceptible to local minima and its performance depends on initialization. Thus as we will see in Sec. IV we average the performance over many Monte-Carlo runs and also study the effect of random initialization as against knowledge-based initialization.

## III. GOALS AND EXPERIMENTAL SETTING

In this paper, we use the dictionary learning methods in the context of period estimation and periodic signal representation using dictionaries. In particular, we seek answers to two questions.

- 1) Can K-SVD learn the periodicity dictionaries like Ramanujan and Farey from the data without explicit periodicity constraints? If so, under what conditions?
- 2) What are the advantages of K-SVD learned dictionaries over known periodicity dictionaries? Does it offer better representation in case of noisy data?

In the remainder of this section, we describe the evaluation metrics and experimental setting.

### A. Evaluation Metrics

In order to compare the learned dictionaries under different settings, we use two evaluation metrics. The dictionary similarity metric  $\rho(\mathbf{D}, \tilde{\mathbf{D}})$  measures how well the columns from the original dictionary  $\mathbf{D}$  are recovered by the learned dictionary  $\tilde{\mathbf{D}}$ . To compute this, for each column in the original dictionary we find a column from the learned dictionary that has the maximum normalized inner product with it. We then take an average of this maximum inner product over all columns of the dictionary. Thus,

$$\rho(\mathbf{D}, \tilde{\mathbf{D}}) = \frac{1}{M} \sum_{i=1}^M \max_k |d_i^H \tilde{d}_k| \quad (12)$$

A value of  $\rho(\mathbf{D}, \tilde{\mathbf{D}})$  close to 1 indicates that most of the columns from the original dictionary are close to some column of the reconstructed dictionary.

We also define two error metrics  $e_1(\mathbf{X}, \mathbf{Y}, \mathbf{D})$  and  $e_2(\mathbf{Y}, \mathbf{D})$  that capture how well the learned dictionary can represent the training data.

$$e_1(\mathbf{X}, \mathbf{Y}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^N \frac{\|\mathbf{D}\gamma_i - \mathbf{x}_i\|_2^2}{\|\mathbf{x}_i\|_2^2} \quad (13)$$

$$e_2(\mathbf{Y}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^N \frac{\|\mathbf{D}\gamma_i - \mathbf{y}_i\|_2^2}{\|\mathbf{y}_i\|_2^2} \quad (14)$$

Here,  $\mathbf{Y}$  is the noisy data matrix,  $\mathbf{X}$  is the noiseless data matrix, and  $\mathbf{D}$  is the learned dictionary. Note that in order to calculate these metrics, we first need to compute the coefficient vectors  $\gamma_i$  from the given noisy data  $\mathbf{Y}$  adhering to sparsity or error constraints. With the coefficients  $\gamma_i$  calculated,  $e_1$  is the relative mean squared error (MSE) of the representation  $\mathbf{D}\Gamma$  with respect to the noiseless data  $\mathbf{X}$ . Note that even if  $\mathbf{Y}$  does not appear directly in the Eq. (13), it is required to find the coefficients  $\gamma_i$ . Since error  $e_1$  captures how well the representation based on the learned dictionary is able to match the noiseless data, it can be regarded as denoising error. Similarly,  $e_2$  is relative MSE with respect to noisy data  $\mathbf{Y}$ . This indicates how well the dictionary can represent noisy data. Note that if the training data does not have any noise then  $\mathbf{X} = \mathbf{Y}$ , and thus  $e_1 = e_2$ .

### B. Experimental Setting

For all our experiments, we first create either a Ramanujan or a Farey dictionary that can represent component periods up to some integer  $P_{max}$ . We then generate  $T_0$ -sparse data from the dictionary and learn a dictionary using K-SVD from this data. The standard (real-valued) K-SVD is used for the real data generated from the Ramanujan dictionary whereas the complex-valued K-SVD is used for the complex data generated from the Farey dictionary.

With the following default values of parameters, we vary one or two parameters at a time and see how the performance varies. The maximum component period  $P_{max}$  is 15 and the signal length or the number of rows of the dictionary is 60. Thus, the number of dictionary columns is  $\Phi(15) = 72$ , so that the true underlying dictionary is overcomplete. We set the sparsity of 5, and 500 data points are used for dictionary learning. 100 Monte Carlo simulations are run for each parameter set, each with 100 K-SVD iterations.

## IV. EXPERIMENTAL RESULTS

Our experiments show that the K-SVD algorithm is indeed able to learn the well-known periodicity dictionaries from data. The accuracy with which this learning takes place (i.e., dictionary similarity and representation errors) depends on a number of parameters in the experiments, as explained in the following subsections.

### A. Variation with SNR and Datapoints

Fig. 1 shows the variation of dictionary similarity with SNR of the training data and the number of datapoints for the Ramanujan dictionary. We see that the K-SVD is able to recover the underlying dictionary when SNR is high, and more data helps recover the dictionary columns even at relatively lower SNR values. Fig. 2 shows similar trends observed for the Farey dictionary. Thus, we see that the complex K-SVD can also learn dictionaries with similar performance.

Figs. 3 and 4 show representation errors  $e_1$  and  $e_2$  for the case of learning the Ramanujan dictionary. Note that both representation errors reduce with higher SNR and more data. However, it is interesting to note that for very few data points,  $e_2$  in fact does not reduce with increasing SNR. Exactly similar trends were also observed for complex K-SVD hence we have skipped those graphs here.

### B. Variation with Sparsity - Fixed and Variable

To see the effect of sparsity, we perform two kinds of experiments. In one experiment, all the training datapoints have fixed sparsity, and in another experiment, different datapoints can have different sparsity - which we call *variable sparse* data. For example, 20 variable sparse data means that the sparsity of different datapoints can be anything between 1 and 20.

Based on Figs. 5 and 6, for both real and complex K-SVD we see that K-SVD learned dictionary has a lower representation error when sparsity is very low or high. On the other hand, the dictionary similarity decreases monotonically as sparsity increases. This suggests that even when we have low similarity to the underlying dictionary, the representation error can be small. Also, note that K-SVD is able to learn a good dictionary even when sparsity is not fixed. In fact, the representation error and dictionary similarity improve in this case. Intuitively, this can be attributed to the fraction of the data which has lesser sparsity as it reveals more information about the columns of the underlying dictionary.

### C. Variation with Number of Allowed Dictionary Columns

Figs. 7 and 8 show the variation with the number of columns the learned dictionary is allowed to have. Note that when the allowed number of columns exceeds 72, we recover all atoms of the underlying dictionary. Also interesting is that if more than 72 columns are allowed, representation error does not reduce but in fact increases.

### D. Variation with Maximum Component Period

From Figs. 9 and 10 we note that with the length fixed, a better dictionary is learned when the data has a lower component period. This means that there needs to be at least a certain number of periods of the periodic components included in the data length. Also, note that the complex-valued K-SVD is much more robust to larger component periods than the real-valued K-SVD. The similarity is relatively higher even for larger periods.

Fig. 1. Variation of Ramanujan dictionary similarity with SNR and datapoints.

Fig. 2. Variation of Farey dictionary similarity with SNR and datapoints.

Fig. 3. Variation of representation error  $e_1$  with SNR and datapoints for Ramanujan dictionary.

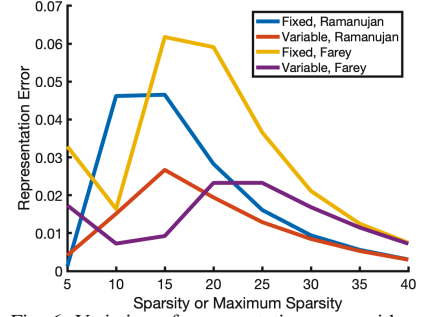
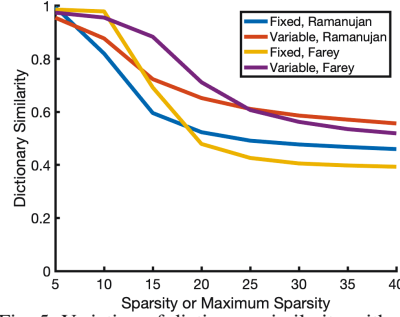


Fig. 4. Variation of representation error  $e_2$  with SNR and datapoints for Ramanujan dictionary.

Fig. 5. Variation of dictionary similarity with sparsity and maximum sparsity.

Fig. 6. Variation of representation error with sparsity and maximum sparsity.

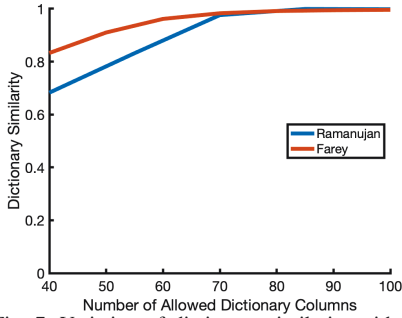


Fig. 7. Variation of dictionary similarity with the allowed number of dictionary columns.

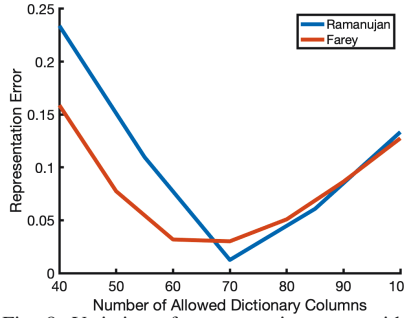


Fig. 8. Variation of representation error with the allowed number of dictionary columns.

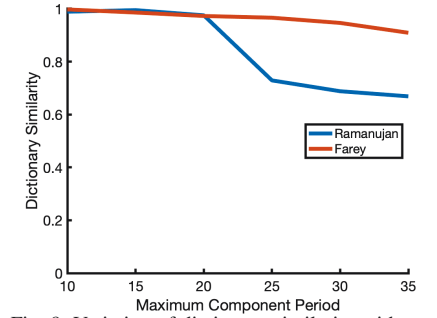


Fig. 9. Variation of dictionary similarity with maximum period.

### E. Variation with Signal Length

Figs. 11 and 12 demonstrate results of similar nature as the signal length is varied. Here, with the maximum period fixed, a longer signal length is preferred, which again emphasizes that there needs to be at least a certain number of periods of the periodic components included in the data length. Here too the complex K-SVD is more robust to smaller signal lengths.

### F. Different Versions and Initializations of K-SVD

Fig. 13 compares two different versions of K-SVD, namely, sparsity constrained formulation (Eq. 8) and error constrained formulation (Eq. 11). For both versions, we consider two initializations - random initialization and Ramanujan initialization. At high SNR, both versions and random as well as knowledge-based initialization of K-SVD perform equally well. Also, notice that for random initialization, error based optimization gives marginally better dictionary similarity than sparsity-based optimization.

Now in pursuit of seeking whether there is any advantage of learned dictionaries over the known analytic dictionaries,

we study the variations of representation error for known and trained dictionaries and with different initializations. From Fig. 14 we see that for denoising type error  $e_1$ , Ramanujan dictionary without any further optimization gives better performance than the learned dictionaries. However, with regards to the representation error  $e_2$  (Fig. 15), the learned dictionaries adapt better to the noisy data. Here too, all the trends were similar for the Farey dictionary as well and hence we have skipped those graphs.

Within the learned dictionaries with two different initializations, we see that Ramanujan initialization has both errors smaller than random initialization. Thus the knowledge-based initialization is better than the random initialization for representing noisy data through a learned dictionary.

### V. CONCLUSION

In this paper, we studied the role of a popular dictionary learning algorithm, K-SVD, in the context of dictionaries for period estimation and periodic signal representation. The experiments demonstrate that K-SVD has the ability to



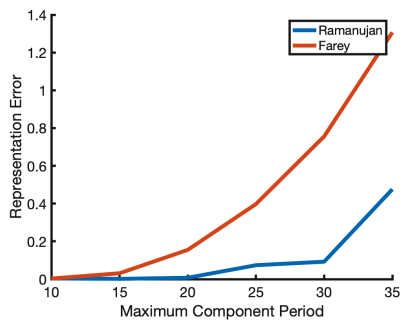


Fig. 10. Variation of representation error with maximum period.

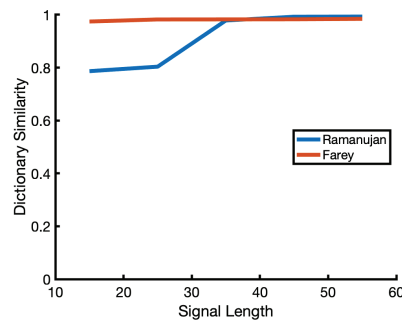


Fig. 11. Variation of dictionary similarity with signal length.

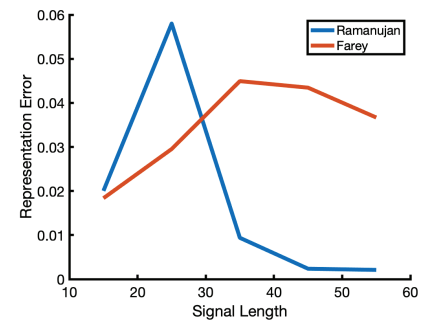


Fig. 12. Variation of representation error with signal length.

Fig. 13. Variation of Ramanujan dictionary similarity for different K-SVD versions and initializations.

Fig. 14. Effect of initialization on representation error  $e_1$  for Ramanujan dictionary.

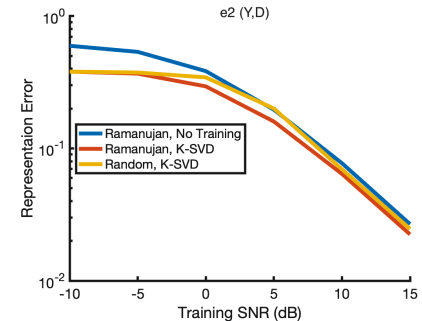


Fig. 15. Effect of initialization on representation error  $e_2$  for Ramanujan dictionary.

learn dictionaries for specialized applications without explicit application-specific constraints. We did not explicitly constrain the dictionary columns to be periodic with different periods, and still, the dictionary could be recovered under favorable circumstances. In some situations, it was also seen that the learned dictionary not being close to ground truth may not imply poor representation and vice versa. The learning algorithm may find an equally good and sparse representation of the data through some other dictionary.

In the future, taking motivation from the results here, we would like to develop a new denoising framework for periodic signals using a hybrid analysis filter bank and synthesis dictionary approach.

#### REFERENCES

- [1] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [2] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.
- [3] N. J. Larsson and A. Moffat, "Off-line dictionary-based compression," *Proceedings of the IEEE*, vol. 88, no. 11, pp. 1722–1732, 2000.
- [4] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.
- [5] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [7] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, vol. 5, 1999, pp. 2443–2446 vol.5.
- [8] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya, "Learning unions of orthonormal bases with thresholded singular value decomposition," in *Proceedings. (ICASSP '05). IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5, 2005, pp. v/293–v/296 Vol. 5.
- [9] M. Yaghoobi, L. Daudet, and M. E. Davies, "Parametric dictionary design for sparse coding," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4800–4810, 2009.
- [10] P. P. Vaidyanathan, "Ramanujan sums in the context of signal processing — Part II: FIR representations and applications," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4158–4172, Aug. 2014.
- [11] S. V. Tenneti and P. P. Vaidyanathan, "Nested periodic matrices and dictionaries: New signal representations for period estimation," *IEEE Trans. on Signal Processing*, vol. 63, no. 14, pp. 3736–3750, Jul. 2015.
- [12] P. P. Vaidyanathan, "Ramanujan sums in the context of signal processing — Part I: Fundamentals," *IEEE Trans. on Signal Processing*, vol. 62, no. 16, pp. 4145–4157, Aug. 2014.
- [13] P. P. Vaidyanathan and S. Tenneti, "Srinivasa Ramanujan and signal-processing problems," *Philosophical Transactions of the Royal Society A*, vol. 378, Issue 2163, Dec. 2019.
- [14] S. Ramanujan, "On certain trigonometrical sums and their applications in the theory of numbers," *Trans. of the Cambridge Phil. Soc.*, vol. XXII, no. 13, pp. 259–276, 1918.
- [15] P. P. Vaidyanathan and P. Pal, "The Farey-dictionary for sparse representation of periodic signals," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 360–364.
- [16] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.
- [17] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44 vol.1.
- [18] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.
- [19] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.
- [20] M. Elad, R. Rubinstein, and M. Zibulevsky, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," *Technical Report - Computer Science, Technion*, Apr. 2008.
- [21] J. S. Turek, M. Elad, and I. Yavneh, "Clutter mitigation in echocardiography using sparse signal separation," *International Journal on Biomedical Imaging*, Apr. 2015.