Learning Fast Sparsifying Overcomplete Dictionaries

Cristian Rusu and John Thompson
Institute for Digital Communications, University of Edinburgh
Email: {c.rusu, john.thompson}@ed.ac.uk

***Abstract*—In this paper we propose a dictionary learning method that builds an overcomplete dictionary that is computationally efficient to manipulate, i.e., sparse approximation algorithms have sub-quadratic computationally complexity. To achieve this we consider two factors (both to be learned from data) in order to design the dictionary: an orthonormal component made up of a fixed number of fast fundamental orthonormal transforms and a sparse component that builds linear combinations of elements from the first, orthonormal component. We show how effective the proposed technique is to encode image data and compare against a previously proposed method from the literature. We expect the current work to contribute to the spread of sparsity and dictionary learning techniques to hardware scenarios where there are hard limits on the computational capabilities and energy consumption of the computer systems.**

I. Introduction

Dictionary learning [1] is a factorization technique introduced in the sparse representation literature [2] that has been successfully applied in research fields like image processing [3] and wireless communications [4]. Given a training dataset with N samples $\mathbf{Y} \in \mathbb{R}^{n\times N}$, the dictionary learning problem can be stated as the NP-hard [5], non-convex optimization problem

$$\begin{aligned} \underset{\mathbf{D},\,\mathbf{X}}{\text{minimize}} \quad & \|\mathbf{Y}-\mathbf{DX}\|_F^2 \\ \text{subject to} \quad & \|\mathbf{d}_j\|_2 = 1,\ 1\le j\le m, \\ & \|\mathbf{x}_i\|_0 \le s,\ 1\le i\le N, \end{aligned} \tag{1}$$

for which several efficient algorithms that have been proven to perform very well in practice have been proposed [6], [7], [8], [9]. The problem is hard because the objective function is bilinear, i.e., both the dictionary $\mathbf{D}$ and the sparse approximation matrix $\mathbf{X}$ are unknown, and the constraints are non-convex, i.e., the ℓ_2 equality constraints on the m columns of the dictionary and the ℓ_0 pseudo-norm constraints on the N columns of $\mathbf{X}$. The size of the dictionary m and the target sparsity s are provided by the user.

It is impossible to enumerate all the algorithms proposed to approximately solve (1) (or problems similar to (1)) since the literature on this subject is already vast. This flurry of research is one of the reasons why dictionary learning (in all its variants, with different numerical characteristics and additional constraints) is seen as such a powerful algorithmic tool.

One downside of constructing a dictionary $\mathbf{D}$ from given data is that, in general, it will lack any specific structure. Classically, transforms (square and usually orthonormal dictionaries), like the Fourier and wavelet, have been extensively used in signal processing due to their internal structure that allows for their use in numerically efficient procedures, i.e., $O(n\log n)$ complexity.

Previous work in the literature has already studied the possibility of building computationally efficient dictionaries learned from data. For example, square dictionaries (orthonormal and general) that are also known as transforms with complexity $O(n\log n)$ have been introduced in [10], [11] and [12]. For more general dictionaries, previous work has explored several ways to introduce structure that improves the computationally complexity of manipulating the dictionary: [13], [14], [15] and [16] to name a few.

In this paper we will introduce a new method to construct computationally efficient overcomplete dictionaries. We extend previous work for learning orthonormal transforms to the non-orthonormal [11] and overcomplete scenarios and compare against a similar method from the literature [16]. Given the sparse representations, we propose a dictionary structure for which we are able to introduce an optimization procedure that has polynomial complexity and that is monotonically decreasing to a local minimum point. We then use image data to show the coding capabilities of the proposed dictionaries and how they compare with a competing, similar, method called the Sparse K-SVD [16] and to the classic K-SVD algorithm [7].

II. The proposed dictionary structure

We consider the so-called double sparsity dictionary model [16], [17] that $\mathbf{D} \in \mathbb{R}^{n\times m}$ has the factorization

$$\mathbf{D} = \mathbf{US}, \tag{2}$$

where we have denoted:

- $\mathbf{U} \in \mathbb{R}^{n\times n}$ is an orthonormal matrix that has a factorization into g G-transforms as $\mathbf{U} = \prod_{k=1}^{g} \mathbf{G}_{i_k j_k}$ [18] such that matrix-vector multiplication between $\mathbf{U}$ and a given vector takes $6g$ operations.
- $\mathbf{S} \in \mathbb{R}^{n\times m}$ is a sparse matrix with $p \ge 1$ non-zero entries in each of its columns (therefore, there are pm non-zero entries in $\mathbf{S}$). The columns have unit ℓ_2 norm – this constraint together with the orthonormal $\mathbf{U}$ ensure that all columns of the dictionary $\mathbf{D}$ are also ℓ_2 normalized, i.e., since $\mathbf{d}_j = \mathbf{U}\mathbf{s}_j$ and therefore $\|\mathbf{d}_j\|_2 = 1$ because $\|\mathbf{s}_j\|_2 = 1$ (as $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$).

ISBN 978-0-9928626-7-1

Unlike the previous approaches [16], [17], our proposed optimization problem for learning computationally efficient dictionaries updates both $\mathbf{U}$ and $\mathbf{S}$ following:

$$\begin{aligned} \underset{\mathbf{U},\,\mathbf{S},\,\mathbf{X}}{\text{minimize}} \quad & \|\mathbf{Y} - \mathbf{USX}\|_F^2 \\ \text{subject to} \quad & \mathbf{U} = \prod_{k=1}^{g} \mathbf{G}_{i_k j_k}, \\ & \|\mathbf{s}_j\|_0 \leq p,\ \|\mathbf{s}_j\|_2 = 1,\ 1 \leq j \leq m, \\ & \|\mathbf{x}_i\|_0 \leq s.\ 1 \leq i \leq N. \end{aligned} \tag{3}$$

The benefit of the proposed method is that the resulting dictionary is computationally efficient to manipulate, i.e., $\mathbf{D}^T\mathbf{y}$ (which is essential to building the sparse approximation of $\mathbf{y}$ in the dictionary $\mathbf{D}$ [8]) takes

$$N_{\text{new}} = 2pm + 6g, \tag{4}$$

operations instead of the classic $N_{\text{old}} = 2nm$. In order to keep the new dictionary efficient (linear computational complexity) we consider parameters $p = O(1)$ and $g = O(n)$, i.e., $p \ll g$ and in fact, in the results section, we keep $p \in \{2, 3\}$ only.

Using this structure of the dictionary we now explain how the components $\mathbf{U}$ and $\mathbf{S}$ and the representations $\mathbf{X}$ are updated.

A. The update of $\mathbf{U}$

We consider that the orthonormal $\mathbf{U}$ has a factorization as

$$\mathbf{U} = \prod_{k=1}^{g} \mathbf{G}_{i_k j_k} = \mathbf{G}_{i_1 j_2} \dots \mathbf{G}_{i_g j_g}, \tag{5}$$

where each $\mathbf{G}_{i_k j_k}$ is a G-transform as defined in [11]:

$$\mathbf{G}_{i_k j_k} = \begin{bmatrix} \mathbf{I}_{i_k - 1} & & & & \\ & * & & * & \\ & & \mathbf{I}_{j_k - i_k - 1} & & \\ & * & & * & \\ & & & & \mathbf{I}_{n - j_k} \end{bmatrix} \in \mathbb{R}^{n \times n}, \tag{6}$$

where we denote the non-trivial part of $\mathbf{G}_{i_k j_k}$ as

$$\tilde{\mathbf{G}}_{i_k j_k} = \left\{ \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}, \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \right\} \in \mathbb{R}^{2 \times 2},\ c_k^2 + s_k^2 = 1. \tag{7}$$

With this structure and with $\mathbf{S}$, $\mathbf{X}$ and $\mathbf{G}_{i_t j_t}$, $t \neq k$ fixed, our goal is to update each $\mathbf{G}_{i_k j_k}$ sequentially by solving

$$\underset{\mathbf{G}_{i_k j_k}}{\text{minimize}} \left\| \prod_{t=k-1}^{1} \mathbf{G}_{i_t j_t}^T \mathbf{Y} - \mathbf{G}_{i_k j_k} \prod_{t=k+1}^{g} \mathbf{G}_{i_t j_t} \mathbf{SX} \right\|_F^2. \tag{8}$$

The objective function is of the form $\|\mathbf{Y}_k - \mathbf{G}_{i_k j_k}\mathbf{X}_k\|_F^2$ and this problem has been studied in [11] where a procedure of finding the optimum $\mathbf{G}_{i_k j_k}$ is described. Briefly, the optimum $\mathbf{G}_{i_k j_k}$ is found by studying the spectral properties of 2×2 sub-matrices of $\mathbf{Z}_k = \mathbf{Y}_k \mathbf{X}_k^T$ to deciding the indices (i_k, j_k) and then solving an orthonormal Procrustes problem [19] for these indices to construct $\tilde{\mathbf{G}}_{i_k j_k}$.

B. The update of $\mathbf{S}$

We consider that the sparse matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ has column structure as

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_m \end{bmatrix}, \tag{9}$$

where for each column $\mathbf{s}_j$ we have the sparsity constraints $\|\mathbf{s}_j\|_0 = p$ and the ℓ_2 normalization $\|\mathbf{s}_j\|_2 = 1$ for $j = 1, \dots, m$. We will use the same idea of sequential updated, this time column by column. With $\mathbf{U}$ and $\mathbf{X}$ fixed, consider the development of the objective function in (3)

$$\begin{aligned} \|\mathbf{Y} - \mathbf{USX}\|_F^2 &= \|\mathbf{U}^T\mathbf{Y} - \mathbf{SX}\|_F^2 \\ &= \|\mathbf{U}^T\mathbf{Y} - \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_m \end{bmatrix} \mathbf{X}\|_F^2 \\ &= \left\| \mathbf{U}^T\mathbf{Y} - \sum_{i=1, i\neq j}^{m} \mathbf{s}_i \mathbf{x}_i^T - \mathbf{s}_j \mathbf{x}_j^T \right\|_F^2 \\ &= \|\mathbf{R}_j - \mathbf{s}_j \mathbf{x}_j^T\|_F^2, \end{aligned} \tag{10}$$

where $\mathbf{x}_i^T$ is the i^{th} row of $\mathbf{X}$ and we have denoted $\mathbf{R}_j$ the residual when updating the j^{th} column of $\mathbf{S}$. Let us denote by $\mathcal{I}_j$ the set of indices of the non-zero entries of $\mathbf{s}_j$ (the support of size p of the column) which is fixed throughout the iterations of the algorithm and with $\mathcal{J}_j$ the set of indices of the non-zero entries on $\mathbf{x}_j^T$ (i.e., the columns of $\mathbf{X}$ that use the j^{th} atom of the dictionary). With this notation we have that

$$\|\mathbf{R}_j - \mathbf{s}_j \mathbf{x}_j^T\|_F^2 = \|\mathbf{R}_{\mathcal{I}_j, \mathcal{J}_j} - \mathbf{s}_{\mathcal{I}_j, j} \mathbf{x}_{j, \mathcal{J}_j}^T\|_F^2, \tag{11}$$

where $\mathbf{s}_{\mathcal{I}_j, j} \in \mathbb{R}^{p \times 1}$, $\mathbf{x}_{j, \mathcal{J}_j}^T \in \mathbb{R}^{1 \times |\mathcal{J}_j|}$ and $\mathbf{R}_{\mathcal{I}_j, \mathcal{J}_j} \in \mathbb{R}^{p \times |\mathcal{J}_j|}$ is the residual matrix restricted only to its non-trivial rows and columns. This is now a rank-1 update problem that can be easily solved via the singular value decomposition. The constraint $\|\mathbf{s}_j\|_2 = 1$ is implicitly obeyed since the singular vector $\mathbf{s}_{\mathcal{I}_j, j}$ has ℓ_2 norm one. This type of update is the regular one used in the K-SVD [7] and Sparse K-SVD [16] algorithms (the rank-1 decomposition restricted by the set $\mathcal{J}_j$) but adapted to the case when the dictionary atom is sparse (and we know the sparsity level p and the location of the non-zero entries). For example, the Sparse K-SVD approach has a more sophisticated update scheme (which basically amounts to an OMP algorithm followed by an ℓ_2 normalization) for each atom in $\mathbf{S}$ because it uses a fixed $\mathbf{U}$ (usually a well known numerically efficient transform like the DCT or wavelet). Since we are updating $\mathbf{U}$ as well, we can fix the sparsity pattern in $\mathbf{S}$ and only update its values. The advantage is that the resulting numerical algorithm performs the singular values decomposition over a variable of size p instead of size n and avoids the OMP procedure to update the columns $\mathbf{s}_j$. Therefore, the update of $\mathbf{S}$ is fast and ensures a monotonic decrease in the objective function of (3).

In the initialization of the proposed method we set and fix the sets $\mathcal{I}_j$ and are careful to ensure that there is an (approximate) uniform distribution of the non-zero indices from $\{1, n\}$, i.e., each index appears approximately pmn^{-1} times.

ISBN 978-0-9928626-7-1 © EURASIP 2017

Algorithm 1 – F–DLA. Fast Dictionary Learning.

Input: The dataset $\mathbf{Y} \in \mathbb{R}^{n \times N}$, the number of G-transforms g in the structure of $\mathbf{U}$, the number of atoms in the dictionary m, the number of non-zero entries p in each column of the sparse component of the dictionary, the target sparsity s and the number of iterations K.

Output: The overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$ factored as $\mathbf{D} = \mathbf{US}$ constrained as in (3) and the sparse representations $\mathbf{X} \in \mathbb{R}^{m \times N}$ such that $\|\mathbf{Y} - \mathbf{USX}\|_F^2$ is reduced.

Initialization:

1) Perform the economy size singular value decomposition of the dataset $\mathbf{Y} = \mathbf{U\Sigma V}^T$ and keep the orthonormal component $\mathbf{U}$.

2) Establish the sets $\mathcal{I}_j$ (the p non-zero indices of each columns $\mathbf{s}_j$ of $\mathbf{S}$) and initialize $\mathbf{s}_j$ on this support from a standard Gaussian distribution. Normalize $\mathbf{s}_j = \mathbf{s}_j \|\mathbf{s}_j\|_2^{-1}$.

3) With $\mathbf{U}$ and $\mathbf{S}$ fixed, compute the sparse representations $\mathbf{X} = \text{OMP}(\mathbf{U}, \mathbf{S}, \mathbf{Y}, s)$.

Iterations $1, \ldots, K$:

1) Update $\mathbf{U}$: with $\mathbf{S}$ and $\mathbf{X}$ fixed, for $k = 1, \ldots, g$ update the new $\mathbf{G}_{i_k j_k}$, with all other transforms fixed, such that (8) is minimized.

2) Update $\mathbf{S}$: with $\mathbf{U}$ and $\mathbf{X}$ fixed, for $j = 1, \ldots, m$ establish the sets $\mathcal{J}_j$ (the indices of the elements in the dataset $\mathbf{Y}$ that use the atom $\mathbf{d}_j$), construct the residual matrix $\mathbf{R}_{\mathcal{I}_j, \mathcal{J}_j}$ and perform a rank-1 factorization using the singular value decomposition to update $\mathbf{s}_j$ and $\mathbf{x}_j^T$ following (11).

3) Update $\mathbf{X}$: with $\mathbf{U}$ and $\mathbf{S}$ fixed, compute the sparse representations $\mathbf{X} = \text{OMP}(\mathbf{U}, \mathbf{S}, \mathbf{Y}, s)$.

C. The update of the sparse representations $\mathbf{X}$

To construct the sparse representations $\mathbf{X} \in \mathbb{R}^{m \times N}$ we use a Batch Orthogonal Matching Pursuit (Batch-OMP) [8] for each element $\mathbf{y}_i,\ i = 1, \ldots, N$ of the dataset. To use the Batch-OMP [8] we have to compute the projections $\mathbf{D}^T\mathbf{Y}$ and the Gram matrix of the dictionary $\mathbf{G} = \mathbf{D}^T\mathbf{D}$. Using the structure we consider for the dictionary we reach that

$$\mathbf{D}^T\mathbf{Y} = \mathbf{S}^T\mathbf{U}^T\mathbf{Y}, \tag{12}$$

which is done efficiently since $\mathbf{U}^T\mathbf{Y}$ takes $6gN$ operations while the multiplication with the sparse matrix $\mathbf{S}^T$ takes $2pmN$ operations and for the Gram matrix we have that

$$\mathbf{G} = \mathbf{D}^T\mathbf{D} = \mathbf{S}^T\mathbf{U}^T\mathbf{US} = \mathbf{S}^T\mathbf{S}, \tag{13}$$

which can be computed in less than $pm(m-1)$ operations, since each column of $\mathbf{S}$ has sparsity p and we only need to compute the upper triangle of $\mathbf{G}$.

D. The initialization

The optimization problem in (3) is non-convex due to both its objective function and the constraints. The optimization sub-problems we defined in the previous sub-section solve for several components in the original problem in a local sense (once component is optimized as the others are kept fixed). Therefore, a good initialization [20] is crucial to ensuring that the whole procedure converges to a solution with a low objective function value. There are three components to discuss: $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{X}$ – but we are concerned mostly with the first two, as $\mathbf{X}$ follows immediately by sparse approximation techniques if the overall dictionary $\mathbf{D}$ is known.

A singular value decomposition of the dataset can produce $\mathbf{Y} = \mathbf{U\Sigma V}^T$ and we can keep the orthonormal basis $\mathbf{U}$ for our purposes. Of course, this matrix does not have an explicit representation in terms of g G-transforms. The sparse matrix $\mathbf{S}$ is initialized randomly. For each p-sparse column $\mathbf{s}_j$ we decide its support $\mathcal{I}_j$ randomly and its initial values drawn from the standard Gaussian distribution. With these two components given, the sparse representations $\mathbf{X}$ follow immediately via the OMP.

Alternative initialization mechanism could use a sparse representation $\mathbf{X}$ given for example by the classic K-SVD algorithm (or other algorithms) and the $\mathbf{U}$ given by the singular value decomposition and based on these optimize over the $\mathbf{S}$. At this point, the quality of the initialization can only be measured experimentally by numerical simulation.

E. Putting it all together: the proposed method

The proposed method, detailed in Algorithm 1, is composed of three blocks: update the orthonormal component $\mathbf{U}$ by updating each G-transform in its structure, update the sparse component $\mathbf{S}$ column by column and update the sparse representations $\mathbf{X}$.

For $p = 1$ and $m = n$ the proposed method reduces to the G_g–DLA algorithm in [11] while for $g = 0$ and $p = n$ it reduces to the K-SVD algorithm [7].

III. Results

In this section we present numerical experimental results on image data to validate the proposed approach. We use image data since in this case we know of computationally efficient transforms (like the DCT [21]) that perform very well in terms of representation performance, i.e., the objective function of (1). To measure this performance we use the relative representation error of the dictionary $\mathbf{D}$ defined as

$$\epsilon = \|\mathbf{Y} - \mathbf{DX}\|_F^2 \|\mathbf{Y}\|_F^{-2}\ (\%). \tag{14}$$

The test dataset $\mathbf{Y} \in \mathbb{R}^{n \times N}$ consists of 8×8 non-overlapping image patches from popular test images often used in the image processing literature (pirate, peppers, boat etc.) with their means removed and normalized $\mathbf{Y} = \mathbf{Y}/255$. We have $n = 64$ and $N = 12288$.

Fig. 1 shows the evolution of the representation error with each iteration. For the Sparse K-SVD approach the convergence is fast (in just a few iterations) while F-DLA (due to the higher number of degrees of freedom) has a slower convergence. We show results for $g = 128$ since for this value the orthonormal component $\mathbf{U}$ has the same computational complexity as the DCT used in the Sparse K-SVD. Then Fig.

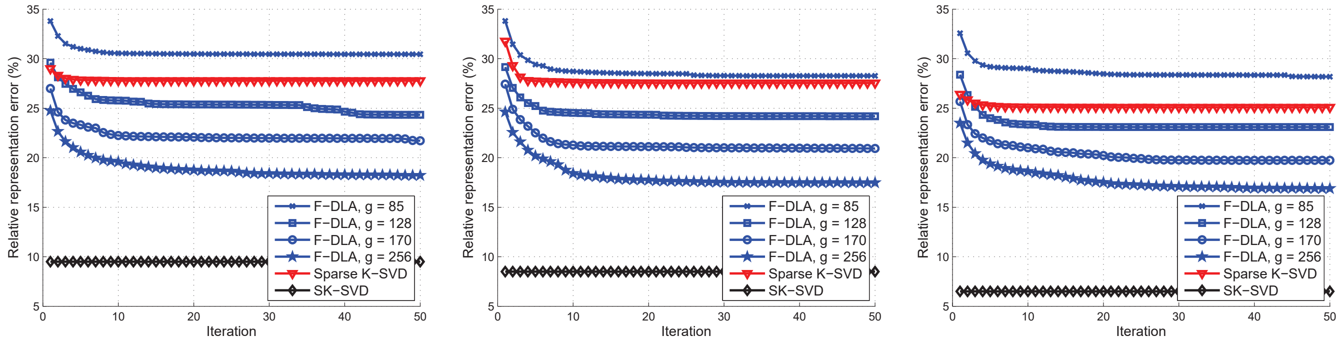

Fig. 1: Evolution of the relative representation error for 50 iterations of the proposed algorithm and the competing Sparse K-SVD approach [16] for a dictionary $\mathbf{D} \in \mathbb{R}^{n\times m}$ with $m \in \{64, 128, 256\}$ (from left to right) for $p = 2$ and $s = 4$. We show the SK-SVD approach [9] as reference.

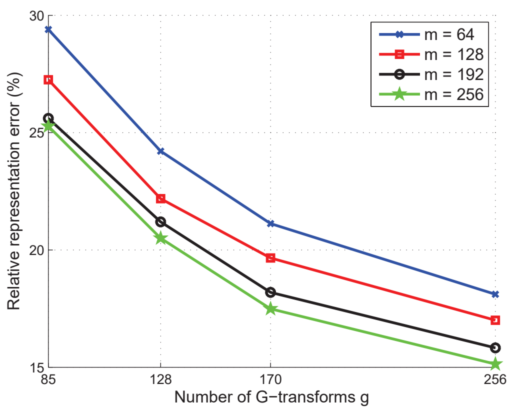

Fig. 2: Relative representation error achieved by dictionaries created via the proposed F-DLA for various m and g for fixed $p = 3$ and $s = 4$.

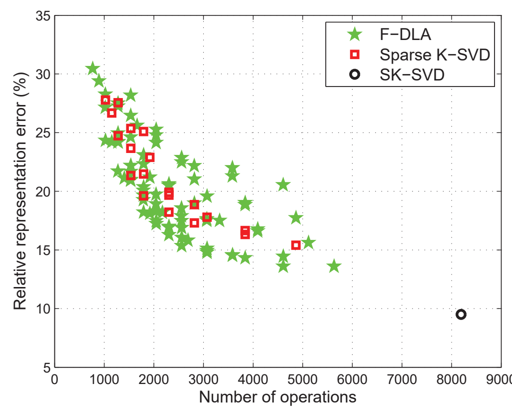

Fig. 3: Relative representation error of the dictionaries $\mathbf{D} \in \mathbb{R}^{n\times m}$ versus their matrix-vector computational complexity $\mathbf{D}^T\mathbf{y}$ when designed via the F-DLA and Sparse K-SVD [16] with the DCT basic dictionary for $m \in \{64, 128, 192, 256\}$ and $g \in \{85, 128, 170, 256\}$. For reference we also show the general dictionary $\mathbf{D} \in \mathbb{R}^{n\times n}$ designed by SK-SVD [9]. Sparsity parameters are set to $s = 4$ and $p \in \{2, 3, 4, 6, 8\}$.

2 shows the representation error against the dictionary size m and number of G-transforms g. The figure shows the trade-off between these two degrees of freedom: for $g = 85$ and $m \in \{192, 256\}$ the representation performance is very similar despite the difference in the number of atoms since the new extra atoms are constrained by the same number of degrees of freedom.

Fig. 3 shows the objective function of the dictionary learning problem against the computational complexity of operating the dictionary in sparse approximation algorithms, i.e., matrix-vector multiplication. F-DLA achieves the best trade-offs when compared with the Sparse K-SVD approach. We observe that the gap between these two approaches, in representation performance, increases with the computational complexity. Also notice that we incur a doubling in computational complexity to close the gap between the representation error of the best dictionaries designed via F-DLA and SK-SVD. For reference we also show the complexity of using a non-structured square dictionary build with SK-SVD.

A general conclusion that might be drawn from Fig. 3 is that the DCT dictionary is well suited for smooth image data and (at least for the cost of $n \log n$ degrees of freedom) no substantial improvement in the representation error can be achieved. We note that a full, general, dictionary $\mathbf{D}$, i.e., full nm degrees of freedom, can further reduce the representation error of course at the cost of full computational complexity.

IV. Conclusions

In this paper we propose a dictionary learning algorithm that builds structured overcomplete dictionaries that are computationally efficient when used in sparse approximation algorithms. We show that the proposed method outperform previous methods from the literature that build structured overcomplete dictionaries on image data. We provide a potential solution to bridge the gap between the classic, computationally efficient, transforms such as the discrete cosine transform and unstructured overcomplete learned dictionaries.

ISBN 978-0-9928626-7-1 © EURASIP 2017

Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014277/1 and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

References

[1] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[2] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Proc.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[4] C. Rusu, R. Mendez-Rial, N. Gonzalez-Prelcic, and R. W. Heath, "Low complexity hybrid sparse precoding and combining in millimeter wave MIMO systems," in *Proc. IEEE ICC*, 2015.

[5] A. M. Tillmann, "On the computational intractability of exact and approximate dictionary learning," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 45–49, 2014.

[6] K. Engan, S. O. Aase, and J. H. Husøy, "Method of optimal directions for frame design," in *Proc. IEEE ICASSP*, 1999, pp. 2443–2446.

[7] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[8] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion*, 2008.

[9] C. Rusu and B. Dumitrescu, "Stagewise K-SVD to design efficient dictionaries for sparse representations," *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 631–634, 2012.

[10] C. Rusu, N. Gonzalez-Prelcic, and R. Heath, "Fast orthonormal sparsifying transforms based on Householder reflectors," *IEEE Trans. Sig. Process.*, vol. 64, no. 24, pp. 6589–6599, 2016.

[11] C. Rusu and J. Thompson, "Learning fast sparsifying transforms," *https://arxiv.org/abs/1611.08230*, 2016.

[12] O. Chabiron, F. Malgouyres, J.-Y. Tourneret, and N. Dobigeon, "Toward fast transform learning," *Technical report*, 2013.

[13] A. B. Lee, B. Nadler, and L. Wasserman, "Treelets - an adaptive multi-scale basis for sparse unordered data," *The Annals of Applied Statistics*, vol. 2, no. 2, pp. 435–471, 2008.

[14] L. L. Magoarou and R. Gribonval, "Learning computationally efficient dictionaries and their implementation as fast transforms," *arXiv:1406.5388*, 2014.

[15] C. Rusu and B. Dumitrescu, "Block orthonormal overcomplete dictionary learning," in *21st European Sig. Proc. Conf.*, 2013, pp. 1–5.

[16] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Sig. Process.*, vol. 58, no. 3, pp. 1553–1564, 2010.

[17] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Imag. Process.*, vol. 22, no. 12, pp. 4598–4612, 2013.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

[19] P. Schonemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[20] C. Rusu and B. Dumitrescu, "An initialization strategy for the dictionary learning problem," in *Proc. IEEE ICASSP*, 2014, pp. 6731–6735.

[21] W.-H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Communications*, vol. 25, no. 9, pp. 1004–1009, 1977.