# Compressed Dictionary Learning

Flavio Teixeira and Karin Schnass
Dept. of Mathematics, University of Innsbruck
Technikerstraße 13, 6020 Innsbruck, Austria
{flavio.teixeira & karin.schnass}@uibk.ac.at

Low complexity models of high-dimensional data lie at the heart of many efficient solutions in modern signal processing. One such model is that of sparsity in a dictionary, where every signal in the data class at hand has a sparse expansion in a predefined basis or frame. In mathematical terms we say that there is a set of $K$ unit-norm vectors $\boldsymbol{\phi}_k \in \mathbb{R}^d$, also referred to as atoms, collected as columns in the dictionary matrix $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_K)$, and that every data point $\boldsymbol{y} \in \mathbb{R}^d$ can be approximately represented as $\boldsymbol{y} \approx \boldsymbol{\Phi}_\mathcal{I} \boldsymbol{x}_\mathcal{I} = \sum_{i \in \mathcal{I}} \boldsymbol{x}(i)\boldsymbol{\phi}_i$, for an index set $\mathcal{I}$ of cardinality $S$ with $S \ll d$.

A fundamental question associated with the sparse model is how to find a suitable dictionary providing sparse representations. When taking a learning rather than a design approach this problem is known as *dictionary learning* or *sparse component analysis*. In its most general form dictionary learning can be seen as a matrix factorization problem. Given a set of $N$ data points in $\mathbb{R}^d$, represented by the $d \times N$ matrix $\boldsymbol{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_N)$, decompose it into a $d \times K$ dictionary $\boldsymbol{\Phi}$ and a set of sparse coefficients, $Y = \boldsymbol{\Phi} X$, where $X$ is sparse.

Since the seminal paper by Olshausen and Field, [1], a myriad of dictionary learning algorithms have been developed and recently also theory on the problem has started to emerge. For an overview of dictionary learning algorithms see [2], while pointers to the main theoretical results can be found in [3]. Despite the recent developments, so far there exist no efficient algorithms with global recovery guarantees, and even the algorithms that are not supported by theoretical results become computationally intractable as the signal dimension increases.

In this paper we take a step towards increasing computational efficiency of dictionary learning, thus making it applicable to high-dimensional data. As a starting point for our development we use the residual version of the Iterative Thresholding and K-Means (ITKM) algorithm presented in [4], which is supported not only by experimental validation but also by local convergence results. Given an initialization dictionary $\boldsymbol{\Psi}$, dictionary learning is carried out by iteratively performing two operations: (1) finding the sparse support $\mathcal{I}_n^t$ of each point in the data set $\boldsymbol{Y}$ by using thresholding as

$$\mathcal{I}_n^t = \arg\max_{|\mathcal{I}|=S} \|\boldsymbol{\Psi}_\mathcal{I}^* \boldsymbol{y}_n\|_1, \tag{1}$$

and (2) updating the dictionary via $K$-residual means. For most admissible sparsity levels which still allow for stable dictionary recovery, the computationally most expensive operation of ITKM is finding the sparse support. This entails the calculation of the matrix product $\boldsymbol{\Psi}^* \boldsymbol{Y}$ of cost $O(dKN)$ at each iteration. Although this is a quite low cost compared to popular dictionary learning algorithms such as the K-SVD algorithm, [5], which additionally requires the calculation of K leading singular vectors in each iteration, learning dictionaries for high-dimensional data can still be prohibitively expensive.

We therefore introduce the Iterative Compressed-Thresholding and K-Means (IcTKM) algorithm for fast dictionary learning, which has significantly reduced computational cost and can efficiently process large data sets. The key modification of the ITKM algorithm is based on a fundamental dimensionality-reduction result due to Johnson and Lindenstrauss (JL) [6]. It states that for any set $\mathcal{X}$ in $\mathbb{R}^d$ with $|\mathcal{X}| = N$, there exists a map $f : \mathbb{R}^d \to \mathbb{R}^m$ with $m = O\left(\delta^{-2} \log N\right)$ and $\delta \in (0, \frac{1}{2})$, such that for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{X}$

$$(1-\delta)\|\boldsymbol{u}-\boldsymbol{v}\|_2^2 \le \|f(\boldsymbol{u}) - f(\boldsymbol{v})\|_2^2 \le (1+\delta)\|\boldsymbol{u}-\boldsymbol{v}\|_2^2. \tag{2}$$

Moreover probabilistic matrix constructions can efficiently realize the low-distortion embedding $f : \mathbb{R}^d \to \mathbb{R}^m$ in (2). Recent developments have focused on improving the computational costs associated with the embedding and providing tighter bounds on the required embedding dimension, [7], [8], [9]. In particular, matrices with the Restricted Isometry Property (RIP), as introduced by Candès and Tao in [10], can realize the embedding $f : \mathbb{R}^d \to \mathbb{R}^m$ in (2) with high probability when their column signs are randomized [9]. In the specific case where these RIP matrices are formed by choosing at random a subset of $m$ rows from an orthogonal (discrete Fourier [11] and Cosine [12]) or circulant [13] matrix, the computational cost associated with embedding the data is $O(d \log d)$, and the required embedding dimension assumes the near-optimal bound $m = O(\max\{\delta^{-1} \log^{\frac{3}{2}} N \log^{\frac{3}{2}} d, \delta^{-2} \log N \log^4 d\})$.

In the proposed IcTKM algorithm, we can reduce the computational cost associated with finding the sparse support $\mathcal{I}_n^t$ in (1) by embedding the entire data set $\boldsymbol{Y}$ and the initialization dictionary $\boldsymbol{\Psi}$ with a fast JL transformation. Let $m \times d$ denote the JL transform matrix, e.g, a partial orthogonal or circulant matrix with randomized column signs; we replace the thresholding operation in (1) with the *compressed-thresholding* operation, which we define as

$$\mathcal{I}_n^{ct} := \arg\max_{|\mathcal{I}|=S} \|\boldsymbol{\Psi}_\mathcal{I}^* \boldsymbol{\Gamma}^* \boldsymbol{\Gamma} \boldsymbol{y}_n\|_1. \tag{3}$$

The computational cost associated with compressed-thresholding reduces to $O(\max\{\delta^{-1} \log^{\frac{3}{2}} N \log^{\frac{3}{2}} d, \delta^{-2} \log N \log^4 d\}KN)$ as opposed to $O(dKN)$ of regular thresholding. Thus the embedding distortion $\delta$ controls the performance improvement of IcTKM over ITKM. We also have the following convergence results, see [14] for details: the number of data points $N$ (sample complexity) required for IcTKM to locally identify a dictionary with high probability is essentially the same as that of ITKM, while the embedding distortion $\delta$ increases the best achievable error $\tilde{\varepsilon}$ and reduces the convergence radius of IcTKM; However increasing the minimally achievable error is largely negligible for high-dimensional data, since the realistically achievable error is determined by the sample size. The reduction of convergence radius is somewhat more disappointing, but as we show in our numerical experiments in Figures 1 and 2 and in Table I, in practice this does not affect the good global convergence behavior and reduced computational cost of IcTKM.

## Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under Grant no. Y760. In addition the computational results presented have been achieved (in part) using the HPC infrastructure LEO of the University of Innsbruck.

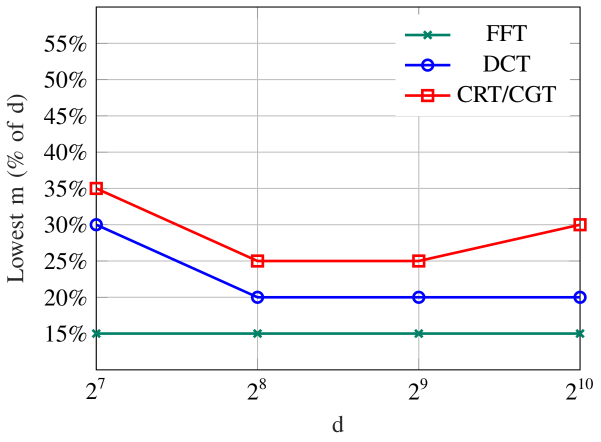

Fig. 1: Lowest achievable embedding dimension to recover 95% of the atoms of the Dirac-DCT dictionary with IcTKM when using random dictionary initialization and the following JL transformations: partial Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Circulant Rademacher Transform (CRT), and Circulant Gaussian Transform (CRT). Training data with $d \in \{2^7, 2^8, 2^9, 2^{10}\}$ and $K = \frac{3}{2}d$ were generated following the signal model in [4] with the parameters: sparsity level $S = \frac{\sqrt{d}}{2}$, noise level $\rho = \frac{1}{2\sqrt{d}}$, and dynamic range varying from 1 to 4.

| JL Transform | $m$ (% of $d$) | No. of Iterations |
|---|---|---|
| FFT | 15% | 65 |
| | 20% | 58 |
| | 30% | 45 |
| | 40% | 39 |
| | 65% | 33 |
| DCT | 20% | 126 |
| | 30% | 52 |
| | 40% | 46 |
| | 65% | 35 |
| CRT/CRT | 30% | 68 |
| | 40% | 55 |
| | 65% | 47 |
| No transf. (ITKM) | 100% | 33 |

TABLE I: Number of iterations required to recover 95% of the atoms from the Dirac-DCT dictionary when using a random dictionary initialization with ITKM, and with IcTKM using different JL transformations. Training data generated with the same parameters as those used in Figure 1 and with $d = 2^{10}$ and $K = \frac{3}{2}d$.

## References

[1] D. Field and B. Olshausen, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[2] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[3] K. Schnass, "A personal introduction to theoretical dictionary learning," *Internationale Mathematische Nachrichten*, vol. 228, pp. 5–15, 2015.

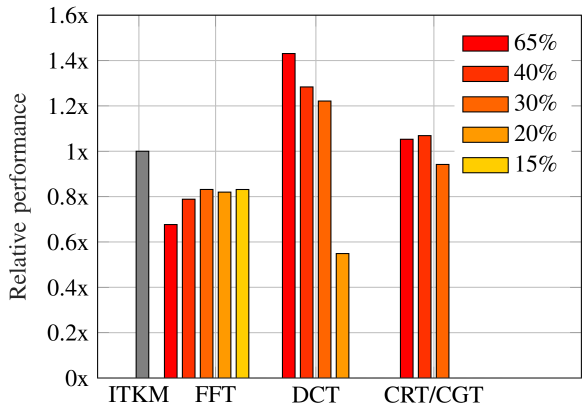

Fig. 2: Relative performance of IcTKM compared to ITKM to process a high-dimensional training data set of $d = 2^{16}$ and $K = \frac{3}{2}\tilde{d}$ with $\tilde{d} = 2^{10}$, and the following parameters: $S = \frac{\sqrt{\tilde{d}}}{2}$, noise level $\rho = \frac{1}{2\sqrt{\tilde{d}}}$, and dynamic range varying from 1 to 4. Best performance in this experiment is achieved by IcTKM with the DCT and embedding dimension 65% of $d$, which is roughly $1.45\times$ faster than ITKM.

[4] ——, "Convergence radius and sample complexity of ITKM algorithms for dictionary learning." *Applied and Computational Harmonic Analysis, to appear*, 2017.

[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, November 2006.

[6] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[7] N. Ailon and E. Liberty, "Fast dimension reduction using Rademacher series on dual BCH codes," *Discrete & Computational Geometry*, vol. 42, no. 4, pp. 615–630, 2008.

[8] N. Ailon and B. Chazelle, "The fast Johnson-Lindenstrauss transform and approximate nearest neighbors," *SIAM J. Comput.*, vol. 39, no. 1, pp. 302–322, May 2009.

[9] F. Krahmer and R. Ward, "New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 3, pp. 1269–1281, 2011.

[10] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[11] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theor.*, vol. 52, no. 12, 2006.

[12] P. Yin, Y. Lou, Q. He, and J. Xin, "Minimization of $\ell_{1-2}$ for compressed sensing," *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. A536–A563, 2015.

[13] H. Rauhut, J. Romberg, and J. A. Tropp, "Restricted isometries for partial random circulant matrices," *Applied and Computational Harmonic Analysis*, vol. 32, no. 2, pp. 242–254, 2012.

[14] K. Schnass and F. Teixeira, "A compressed thresholding and k-means algorithm for fast dictionary learning," *in preparation*.