Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Linear Algebra and its Applications 416 (2006) 48–67

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them

Michal Aharon*, Michael Elad, Alfred M. Bruckstein

Department of Computer Science, Technion—Israel Institute of Technology, Technion City, Haifa 32000, Israel

Received 15 May 2005; accepted 21 June 2005
Available online 26 August 2005
Submitted by A. Berman

Abstract

A full-rank under-determined linear system of equations $\mathbf{Ax} = \mathbf{b}$ has in general infinitely many possible solutions. In recent years there is a growing interest in the sparsest solution of this equation—the one with the fewest non-zero entries, measured by $\|\mathbf{x}\|_0$. Such solutions find applications in signal and image processing, where the topic is typically referred to as "sparse representation". Considering the columns of $\mathbf{A}$ as atoms of a dictionary, it is assumed that a given signal $\mathbf{b}$ is a linear composition of few such atoms. Recent work established that if the desired solution $\mathbf{x}$ is sparse enough, uniqueness of such a result is guaranteed. Also, pursuit algorithms, approximation solvers for the above problem, are guaranteed to succeed in finding this solution.

Armed with these recent results, the problem can be reversed, and formed as an implied matrix factorization problem: Given a set of vectors $\{\mathbf{b}_i\}$, known to emerge from such sparse constructions, $\mathbf{Ax}_i = \mathbf{b}_i$, with sufficiently sparse representations $\mathbf{x}_i$, we seek the matrix $\mathbf{A}$. In this paper we present both theoretical and algorithmic studies of this problem. We establish the uniqueness of the dictionary $\mathbf{A}$, depending on the quantity and nature of the set $\{\mathbf{b}_i\}$, and the sparsity of $\{\mathbf{x}_i\}$. We also describe a recently developed algorithm, the K-SVD, that practically find the matrix $\mathbf{A}$, in a manner similar to the K-Means algorithm. Finally, we demonstrate this algorithm on several stylized applications in image processing.

* Corresponding author.
E-mail address: michalo@cs.technion.ac.il (M. Aharon).

0024-3795/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2005.06.035

Keywords: K-Means; Vector quantization; Codebook; Uniqueness; K-SVD; Training; Dictionary; Atom decomposition; Sparse representation; Basis pursuit; Matching pursuit

1. Introduction

Let $\mathbf{A} \in \mathbb{R}^{n\times k}$ be a full rank matrix with more columns than rows ($n < k$), and $\mathbf{b} \in \mathbb{R}^n$ a corresponding vector. We assume hereafter that the columns of $\mathbf{A}$ are normalized. The under-determined linear system of equations $\mathbf{Ax} = \mathbf{b}$ has infinitely many possible solutions $\mathbf{x} \in \mathbb{R}^k$. In recent years there is a growing interest in finding the sparsest solution of this equation,

$$(P_0) \quad \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to } \mathbf{Ax} = \mathbf{b}, \tag{1}$$

where $\|\cdot\|_0$ is the l^0 semi-norm, counting the non-zero entries of a vector. Another interesting and related question concerns an approximation of the vector $\mathbf{b}$ using a linear combination of few columns of $\mathbf{A}$, by solving

$$(P_{0,\epsilon}) \quad \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to } \|\mathbf{Ax} - \mathbf{b}\|_2 \leqslant \epsilon \tag{2}$$

for a known ϵ.

Solving (P_0) or $(P_{0,\epsilon})$ find applications in signal and image processing, where the topic is typically referred to as "sparse representations". Considering the vector $\mathbf{b}$ as a signal, and the columns of $\mathbf{A}$ as atoms of a dictionary, the solution of (1) leads to the sparsest decomposition of the signal under this dictionary, which stands for an efficient overcomplete signal transform. Solving (2) is even more appealing, as those approximated representations allow sparser and more compact signal description, and are often efficient in removing noise from the signal.

Exact determination of the solution $\mathbf{x}$ for both (1) and (2) proves to be an NP-hard problem [1]. Therefore, approximating algorithms are often used. Indeed, in recent years several very efficient such pursuit algorithms were presented for these problems [2–7], along with an analysis of their behavior. Recent works established that if the sought solution for (1) is sparse enough, this solution is unique, and pursuit techniques succeed in recovering it [8–13]. Later work considered the approximated version $(P_{0,\epsilon})$ and established similarly that sparsity guarantees a stability in the recovery of the solution [14,15]. The most recent activity in this field revisits those questions from a probabilistic standpoint, obtaining more realistic assessments on uniqueness of solutions, and success rates of pursuit algorithms [16–19]. In all these works, the properties of the matrix $\mathbf{A}$ set the limits on the sparsity that consequently ensures a successful approximation with those algorithms. More on the uniqueness properties of sparse representations, the pursuit algorithms, and related theoretical results is briefly surveyed in Section 2.

As we have already mentioned, sparse representations find applications in signal and image processing. Compression, regularization in inverse problems, feature

extraction, and more, are applications that benefit from the sparsity and overcompleteness concepts, as presented above. Indeed, the success of the JPEG2000 compression method can be attributed to the sparsity of the wavelet coefficients of natural images [20]. In denoising, shift invariant wavelet methods exploit overcomplete wavelet representations, and are among the most effective known algorithms for this task [21–24]. Sparsity and overcompleteness have been successfully used for dynamic range compression in images [25], separation of texture and cartoon content in images [26,27], inpainting [28], and more.

In all the above-mentioned works, a pre-specified dictionary matrix $\mathbf{A}$ is chosen and used. An overcomplete wavelet, curvelet, contourlet, steerable wavelet, short-time-Fourier transform, and more, are typically such choices for dictionaries. The natural question is whether one can find the best dictionary for a signal family in mind. As we show next, this turns into a beautiful matrix factorization problem that can be handled successfully.

Consider the following problem: We are given a sufficiently large set of signals $\{\mathbf{b}_i\}_{i=1}^N$, each known to be composed as a sparse linear combination over the full rank matrix $\mathbf{A} \in \mathbb{R}^{n\times k}$. I.e.,

$$\forall 1 \leqslant i \leqslant N, \quad \exists \mathbf{x}_i \text{ such that } \mathbf{b}_i = \mathbf{A}\mathbf{x}_i \ \wedge \ \|\mathbf{x}_i\|_0 \leqslant L \tag{3}$$

for a known cardinality $L \ll n$. Arranging all signals as columns in a matrix $\mathbf{B} \in \mathbb{R}^{n\times N}$, and similarly gathering the coefficient vectors as columns of $\mathbf{X} \in \mathbb{R}^{k\times N}$, we have

$$\mathbf{B} = \mathbf{A}\mathbf{X}. \tag{4}$$

Given the matrix $\mathbf{B}$, we are interested in factorizing it into an arbitrary matrix $\mathbf{A}$ with normalized columns, and a sparse matrix $\mathbf{X}$ with no more than L non-zeros in each column.

In this work we address this factorization problem from a theoretical and algorithmic points of view. First, we consider the question of uniqueness of such factorizations $\mathbf{B} = \mathbf{A}\mathbf{X}$. We prove that under some conditions on the matrices $\mathbf{B}$, $\mathbf{A}$ and $\mathbf{X}$, uniqueness is guaranteed up to a trivial permutation and sign-changes of the columns of $\mathbf{A}$. These conditions refer to the number of columns in $\mathbf{B}$ (i.e., the number of given signals), the cardinality of the columns in $\mathbf{X}$, and specific properties of the desired dictionary $\mathbf{A}$. Section 4 of this paper is devoted to this uniqueness theorem and its proof.

A second question that arises with regard to the discussed factorization concerns the practical ways to achieve it. Interestingly, this problem has been already studied by several researchers in the context of the study of the human visual system [29–32], and signal processing in general [33–35]. In this paper we briefly review these works in Section 3. We then present a recently developed algorithm for this factorization—the K-SVD algorithm—that outperforms its predecessors. Section 5 is devoted to the description of the K-SVD algorithm and its relation to the classical K-Means clustering method. We then show several applications on real images in Section 6 that use the K-SVD and demonstrate its superiority.

Note: Throughout this paper we refer to matrices over the real field ($\mathbb{R}$). A generalization to complex numbers exists, but as our interest is in the analysis of real signals, we choose not to undertake this wider view.

2. Prior art—Sparse representations

2.1. Preliminaries

In order to discuss sparse representations, we first review some basic definitions and facts commonly known in this field. Those definitions are used throughout this paper.

Recall that we refer to the columns of the matrix $\mathbf{A}$ as *atoms* that are later used as building blocks in the construction of $\mathbf{b}$. Such a matrix that holds the atoms as its columns is referred to as a '*dictionary*', and the order of its columns, as also their multiplication by a scalar, is irrelevant. As already said, we often assume that all atoms are normalized. Thus, in our terminology, given any dictionary $\mathbf{A}$, its right-multiplication by a permutation matrix and/or its right-multiplication by a diagonal non-singular matrix, lead to an equivalent dictionary.

The *mutual coherence* of a dictionary $\mathbf{A}$, denoted by $\mu(\mathbf{A})$, is defined as the maximal absolute scalar product between two different normalized atoms of $\mathbf{A}$,

$$\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\mathbf{a}_i^{\mathrm{T}} \mathbf{a}_j|}{\mathbf{a}_i^{\mathrm{T}} \mathbf{a}_i \mathbf{a}_j^{\mathrm{T}} \mathbf{a}_j}. \tag{5}$$

The mutual coherence of a dictionary measures the similarity between the dictionary's atoms. For an orthogonal matrix $\mathbf{A}$, $\mu(\mathbf{A}) = 0$. For an overcomplete matrix ($k > n$) we necessarily have $\mu(\mathbf{A}) > 0$. As we shall see next, there is an interest in dictionaries with as small as possible $\mu\{\mathbf{A}\}$ for sparse representation purposes. If $\mu(\mathbf{A}) = 1$, it implies the existence of two parallel atoms, and this causes confusion in the construction of sparse atom compositions. In [36] it was shown that for a full rank dictionary of size $n \times k$

$$\mu \geqslant \sqrt{\frac{k-n}{n(k-1)}} \tag{6}$$

and equality is obtained for a family of dictionaries called Grassmannian frames. For $k \gg n$ the mutual coherence we can expect to have is thus of the order of $1/\sqrt{n}$.

The *spark* of a dictionary $\mathbf{A}$ is the smallest number of columns that form a linearly dependent set [10]. In-spite the similar definition, note that spark is markedly different from the matrix rank, being the greatest number of linearly independent columns. A trivial relation between the spark $\sigma\{\mathbf{A}\}$ and the mutual coherence $\mu\{\mathbf{A}\}$ is [10]

$$\sigma\{\mathbf{A}\} \geqslant 1 + \frac{1}{\mu\{\mathbf{A}\}}. \tag{7}$$

Referring to the problem posed in (1), we first quote a result that poses a condition on its solution $\mathbf{x}$ such that it guarantees uniqueness:

Theorem 1 (see [10]). *A linear representation over m atoms* (*i.e.*, $\|\mathbf{x}\|_0 = \mathbf{m}$) *is unique if*

$$m < \frac{\sigma\{\mathbf{A}\}}{2}. \tag{8}$$

The work in [14] generalizes this theorem for the solution of (2). It shows that, while exact uniqueness cannot be guaranteed, an approximate one that allows a bounded deviation can be claimed.

These uniqueness theorems imply that no other solution can be found with the same or better cardinality for these problems. Still, the question of how to find such a solution remains unanswered, which leads us to the discussion on pursuit methods.

2.2. Pursuit algorithms

Of course, there is hardly any practical interest in problems that cannot be solved, as seems to be the case for the problems (1) and (2), due to their combinatorial nature. An extensive research in this area in the past years, led to the development of both algorithmic tools and theoretical results, enabling us to refer to those problems as practically solvable, and usable in actual applications. Here we briefly review these main algorithms and results.

The simplest approximating algorithms are the *matching pursuit* (MP) [3] and the *orthogonal matching pursuit* (OMP) algorithms [2,4,5,13]. These are greedy algorithms that select the indices of the non-zero elements sequentially. At each stage, the atom with the largest inner product with the residual signal is chosen. In OMP, an additional projection of the original signal on the already selected atoms is deployed. Similar methods baring the names *step-wise regression*, *pure greedy algorithm*, and more, are manifestations of the same ideas in different fields. Both (1) and (2) are easily addressed by changing the stopping rule of the algorithm.

A second well known pursuit approach is the *basis pursuit* (BP) [6]. It suggests a convexification of the problems posed in (1) and (2), by replacing the ℓ^0 semi-norm with an ℓ^1 norm. The focal under-determined system solver (FOCUSS) is very similar, using the ℓ^p semi-norm with $p \leqslant 1$, as a replacement to the ℓ^0 semi-norm [7,37–39]. Here, for $p < 1$ the similarity to the true sparsity measure is better, but the overall problem becomes non-convex, giving rise to local minima that may mislead in the search for solutions. Lagrange multipliers are used to convert the constraint into a penalty term, and an iterative method is derived based on the idea of iterated reweighed least-squares that handles the ℓ^p semi-norm as an ℓ^2 weighted one.

Both the BP and the FOCUSS can be motivated as *maximum a posteriori* (MAP) estimation techniques, and indeed several works used this reasoning directly [29–32]. MAP estimation can be used to estimate the coefficients as random variables,

by maximizing the posterior $P(\mathbf{x}|\mathbf{y},\mathbf{A}) \propto P(\mathbf{y}|\mathbf{A},\mathbf{x})P(\mathbf{x})$. The prior distribution of the coefficient vector $\mathbf{x}$ is assumed to be a super-Gaussian iid distribution that favors sparsity. For the Laplace distribution this approach is equivalent to BP [31]. A more thorough summary on pursuit algorithms can be found in [40].

2.3. Theoretical results

The uniqueness theorem mentioned above implies that a representation with less than $\sigma\{\mathbf{A}\}/2$ non-zero elements is the sparsest available one, and therefore, if a pursuit algorithm retrieves such a solution it is necessarily the one desired. However, in which cases may we expect the pursuit algorithms to retrieve this exact solution, and when can we guarantee its success? Those kind of questions, concerning the connection between the pursuit algorithm's results and the true solutions to (1) or (2) has been studied extensively. The following is a central result on the expected behavior of the MP and BP methods for sparse enough solutions:

Theorem 2 (see [10–13]). *If the sought solution*, $\mathbf{x}$, *for the problem* (1), *satisfies*

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu\{\mathbf{A}\}}\right) \tag{9}$$

then both BP *and* MP *will recover it exactly*.

This result suggest that for representations with less than $\mathcal{O}(\sqrt{n})$ non-zeros, pursuit methods can succeed. Results of similar nature and strength were developed for structured dictionaries, constructed as a concatenation of several unitary matrices [9,11,41]. Such a dictionary structure is restrictive, but decomposition under this kind of dictionaries can be done efficiently using the block coordinate relaxation (BCR) method [42], being a fast variant of the BP.

Other work considered the approximated version (2) and shown stability in recovery of $\mathbf{x}$ [14,15,41], meaning that pursuit algorithms lead to a solution in the proximity of the true optimal one.

All the above results consider the worst case scenario, and the bounds derived are therefore too pessimistic. The pursuit algorithms are known (empirically) to succeed in recovering sparse representations even when the number of non-zero elements is substantially beyond those bounds. Indeed, the recent front of theoretical activity in this field revisits the above questions from a probabilistic point of view, obtaining more realistic assessments on pursuit algorithms performance and success [16–19]. These works show that even for $\mathcal{O}(n)$ non-zeros[1] in the representations, pursuit methods are expected to succeed with probability one.

[1] And a proper, somewhat small, constant.

3. Prior art—Seeking the dictionary

We now turn to briefly describe the work done on the matrix factorization problem presented earlier. We are given a matrix $\mathbf{B} \in \mathbb{R}^{n\times N}$ ($n \ll N$), containing N signals as its columns. We assume that $\mathbf{B}$ is composed as the multiplication $\mathbf{B} = \mathbf{AX}$, where $\mathbf{A} \in \mathbb{R}^{n\times k}$ ($n < k \ll N$), and $\mathbf{X} \in \mathbb{R}^{k\times N}$ is sparse, such that no more than L non-zeros are found in each of its columns. Given such $\mathbf{B}$, we desire a factorization that retrieves $\mathbf{A}$ and $\mathbf{X}$. Obviously, an arbitrary matrix $\mathbf{B}$ does not necessarily admit such a factorization. Thus, this problem can be rephrased, considering the nearest factorization (using $\|\mathbf{B} - \mathbf{AX}\|_{\mathrm{F}}^2$ or similar measures) that satisfy the sparsity and the dimensions conditions.

In the language of signal processing, this problem is often referred to as finding the optimal overcomplete dictionary for sparse representation. The matrix $\mathbf{B}$ contains a large set of training signals, for which the dictionary is supposed to enable sparse representations. The methods that have been proposed so far consider this problem as an estimation problem, taking into account the factorization error as an additive white noise. A more thorough summary on these method can be found in [40].

The pioneering work by Field and Olshausen suggested to use a maximum-likelihood (ML) based approach [29]. They tried to maximize the conditional probability of the signals $\mathbf{B}$ with respect to the desired dictionary $\mathbf{A}$,

$$\max_{\mathbf{A}} P(\mathbf{B}|\mathbf{A}). \tag{10}$$

Two assumptions are required in order to proceed: First, it is assumed that the training signals are drawn independently, thus providing

$$P(\mathbf{B}|\mathbf{A}) = \prod_{i=1}^{N} P(\mathbf{b}_i|\mathbf{A}). \tag{11}$$

Secondly, the formulation should be using the so far "hidden variable" $\mathbf{x}$. It is incorporated to the likelihood function using the relation

$$P(\mathbf{b}_i|\mathbf{A}) = \int P(\mathbf{b}_i, \mathbf{x}|\mathbf{A})\,\mathrm{d}\mathbf{x} = \int P(\mathbf{b}_i|\mathbf{x}, \mathbf{A}) \cdot P(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{12}$$

From this expression it is implied that given the dictionary $\mathbf{A}$ and a specific training signal $\mathbf{b}_i$, the representation $\mathbf{x}_i$ that satisfies $\mathbf{b}_i = \mathbf{Ax}_i$ is not the sparsest deterministically, but rather probabilistically. The prior distribution of the representation vector $\mathbf{x}$ is assumed to be such that the entries of $\mathbf{x}$ are zero-mean iid, with a super Gaussian distribution (the Cauchy function [30] or the Laplace [29,31]), promoting (but not enforcing) sparsity. An integration over the vector $\mathbf{x}$ is required, which is difficult. Indeed, Olshausen and Field [29] handled this by replacing it with the extremal value of $P(\mathbf{b}_i, \mathbf{x}|\mathbf{A})$. This leads to a minimization problem of the form

$$\min_{\mathbf{A},\{\mathbf{x}_i\}_{i=1}^N} \sum_{i=1}^{N} \|\mathbf{b}_i - \mathbf{Ax}_i\|_2^2 + \lambda \sum_{i=1}^{N} \rho(\mathbf{x}_i). \tag{13}$$

Here $\rho(\cdot)$ stands for a the $-\log$ summation of the super-Gaussian distribution assumed earlier.

In a continuing work, Lewicki, Olshausen and Sejnowski suggested to approximate this integral over the distribution by a Gaussian [30–32], and thus provided a more reliable evaluation of it.

Other important contributions on training of dictionaries for sparse representations has been made by the creators of the FOCUSS algorithm, Rao and Kreutz–Delgado, together with Engan [33,34]. In their work, they pointed out the connection between the sparse coding dictionary design and the vector quantization problem, and proposed some type of generalization of the well known K-Means algorithm. They suggested an iterative algorithm that switches between two main stages—A sparse coding stage, in which sparse representations under a fixed dictionary are found, and dictionary update stage, in which the dictionary is changed in order to better represent the signals, given their known representations. As we shall see next in Section 5, we present a similar approach of the same flavor, but one that is faster due to a boosted update of the dictionary along-side with the representation coefficients.

A very recent work on the topic of dictionary training is the work by Lesage et al. [35]. They suggest a training algorithm for a dictionary composed as a union of ortho-bases [35]. The benefit of such structure is the fact that the obtained dictionary is a tight frame. However, in forcing this structure, one necessarily loses performance.

None of the above works considered the problem as a matrix factorization one, and none considered the consequent questions regarding the uniqueness of such decomposition. We now turn to discuss those issues in depth.

4. Uniqueness of the factorization

We consider first the factorization problem, namely, given the matrix $\mathbf{B} \in \mathbb{R}^{n\times N}$ ($n \ll N$) known to be formed as $\mathbf{B} = \mathbf{AX}$, we desire a factorization that retrieves $\mathbf{A}$ and $\mathbf{X}$, under the conditions we have already specified earlier. Our focus in this section is whether such decomposition is unique. We did not find any explicit reference to this uniqueness question, and yet such uniqueness is implicitly assumed in previous works (e.g., the synthetic experiments in [39,40]).

We now pose a set of assumptions that are mandatory for the uniqueness result we are about to prove. We assume the following:

1. *Support*: The support of all representation vectors in $\mathbf{X}$ (its columns) satisfy

$$\forall 1 \leqslant i \leqslant N, \quad \|\mathbf{x}_i\|_0 = L < \frac{\sigma\{\mathbf{A}\}}{2}. \tag{14}$$

Furthermore, we assume that L is known. Both the knowledge of L, and the fact that the representations are assumed to have exactly L non-zeros (and not less)

can be easily relaxed. Still, these assumptions are posed for the sake of simplicity of the proof.

2. *Richness*: The set of examples in $\mathbf{B}$ includes at least $L+1$ signals for every possible combination of L dictionary elements from $\mathbf{A}$. Thus, we assume that $\mathbf{B}$ includes at least $(L+1)\binom{k}{L}$ signals. This assumption will later be relaxed, and again, it is posed mainly for simplifying the later analysis.
3. *Non-degeneracy*: Given a group of $L+1$ signals that are built of the same L atoms, their rank is expected to be L or less. We assume that any such set leads to a rank L and not less. Similarly, for any group of $L+1$ signals that are built of different support, we assume that the rank of such set is necessarily $L+1$. Both these assumptions mean that no degeneracies in the construction of the signals is allowed. These requirements, although seem complicated, only reflects the fact that there are no degenerate 'coincidences' in the signals construction. Analogously, considering the signals to be generated with L randomly chosen coefficients in $\mathbf{X}$, these degeneracies are of zero probability.

Based on these assumptions we have the following result:

Theorem 3. *Under the above assumptions, the factorization of* $\mathbf{B}$ *is unique, i.e., the factorization* $\mathbf{B}=\mathbf{AX}$ *for which* (i) $\mathbf{A}\in\mathbb{R}^{n\times k}$ *with normalized columns; and* (ii) $\mathbf{X}\in\mathbb{R}^{k\times N}$ *with* L *non-zeros in each column, is unique. This uniqueness is up to a right-multiplication of* $\mathbf{A}$ *by a signed permutation matrix, which does not change the desired properties of* $\mathbf{A}$ *and* $\mathbf{X}$.

Proof. The proof we provide is constructive (although far from being a practical method to deploy in practice), leading to a pair $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{X}}$. Clearly, it is sufficient to prove an equivalence between $\widehat{\mathbf{A}}$ and $\mathbf{A}$ (up to a signed permutation) to guarantee a unique factorization. Given the vector $\mathbf{b}_i$ and a dictionary $\mathbf{A}$, our assumptions imply that a solution to $\mathbf{b}_i=\mathbf{A}\mathbf{x}_i$ exists with exactly L non-zeros. Since $L<\frac{\sigma\{\mathbf{A}\}}{2}$, this is the sparsest possible solution and as such it is unique, due to Theorem 1. Thus, having found $\mathbf{A}$, solving the problem (P_0) we necessarily recover the original $\mathbf{X}$. Permutations and sign changes do not impact this property, and only change the locations and signs of the entries in $\mathbf{X}$, to match the columns in $\mathbf{A}$. Thus, it is sufficient to consider the relation between the original dictionary $\mathbf{A}$ and the recovered one, $\widehat{\mathbf{A}}$.

Next, we describe a coherent process with three stages that leads to $\widehat{\mathbf{A}}$, and show that it necessarily matches the original $\mathbf{A}$. The basic steps of this process are (i) Divide the columns of $\mathbf{B}$ into $J=\binom{k}{L}$ sets—$\{G_1,G_2,\ldots,G_J\}$—each includes all the signals that share the same support (i.e., use the same L atoms from $\mathbf{A}$, denoted as Ω_j for $j=1,2,\ldots,J$); (ii) Detect pairs of sets G_i and G_j that share exactly one mutual atom; and (iii) Extract this mutual atom and form $\widehat{\mathbf{A}}$, which necessarily matches the original $\mathbf{A}$. We now turn to expand on each of those steps.

Stage 1: Clustering the signals

Due to our earlier support assumption, $L < \sigma\{\mathbf{A}\}/2$, and the definition of the spark, every group of L atoms from $\mathbf{A}$ is necessarily linearly independent and as such spans an L-dimensional subspace. In this stage we identify those $\binom{k}{L}$ subspaces, and divide the signals (columns) in $\mathbf{B}$ according to their embedding subspaces.

The clustering of the columns of $\mathbf{B}$ can be done by first testing the rank of all $\binom{k}{L+1} = (k-L)\cdot J/(L+1) > J$ groups of $L+1$-tuples from $\mathbf{B}$. If the rank of such group is $L+1$, it implies that this group of signals do not belong to the same subspace, and as such it is discarded from further consideration. If the rank equals L, it means we have found $L+1$ signals that belong to one of the subspaces related to a set of atoms S_j. The richness assumption assures that such $L+1$ signals exist per each of the J subspaces, and the non-degeneracy assumption prevents a group of $L+1$ signals that do not belong to the same subspace to give a rank L, and impose as a feasible subspace. Thus, we expect to detect exactly J such successful sets of signals from $\mathbf{B}$, and those, denoted by $\{G_j\}_{j=1}^{J}$, will serve as the seeds for the overall clustering. Note that the non-degeneracy assumption also implies that the rank of an arbitrary group of $L+1$ columns cannot be smaller than L.

Having found the seed for each of J groups, $\{G_j\}_{j=1}^{J}$, we now sweep through all the columns in $\mathbf{B}$ that are not assigned yet, and combine them to each of these J sets, testing again the rank. Due to the non-degeneracy assumption, only one group will give a rank L, implying that the tested column belongs to this subspace. All other tests necessarily lead to a rank $L+1$.

Using the above procedure, we divide all signals in $\mathbf{B}$ into $\binom{k}{L}$ sets, each includes signals that are generated by the same group of L atoms in the original dictionary $\mathbf{A}$. The support and richness assumptions ensures that each such subspace will eventually be identified (when testing its corresponding $L+1$ signals). The non-degeneracy assumption ensures that no $L+1$ signals will be mapped into the same set if they were not initially generated by the same L atoms.

Stage 2: Detecting pairs with mutual atom

Given the J sets of signals $\{G_j\}_{j=1}^{J}$, we now test the rank of all $J(J-1)/2$ merged pairs (order plays no role). Every two such merged groups, G_{j_1} and G_{j_2}, are leaning on two sets of atoms from $\mathbf{A}$, Ω_{j_1} and Ω_{j_2}. If the intersection between these two sets is empty, $|\Omega_{j_1} \cap \Omega_{j_2}| = 0$, then the rank is necessarily $2L$. It cannot be higher as each of the sets have a rank L, and cannot be smaller due to the non-degeneracy assumption and the fact that $2L < \sigma\{\mathbf{A}\}$. If the intersection includes one atom, $|\Omega_{j_1} \cap \Omega_{j_2}| = 1$, the rank is necessarily $2L-1$ for the same reasons. Getting a rank smaller than $2L-1$ means a larger intersection. Thus, we take only those pairs that lead to rank $2L-1$.

Let us consider the first atom as the single interaction between pairs $\{G_{j_1}, G_{j_2}\}$. How many such pairs will be found? Putting this atom aside, we remain with $k-1$ atoms from which we have to choose $2L-2$ atoms to participate in the construction of the two sets. These should be divided to two groups of $L-1$ atoms each, and only half count (order, as before, is redundant). Therefore, we have $0.5 \cdot \binom{k-1}{2L-2}\binom{2L-2}{L-1}$ such pairs that intersect only on the first atom. Thus, every atom in the original dictionary $\mathbf{A}$ can and will be found many times, calling for a pruning. Both the evaluation of this atom from the intersection and the pruning are parts of the next and last stage.

Stage 3: Extracting the mutual atom

Assume that we have the pair of groups $\{G_{j_1}, G_{j_2}\}$ known to intersect on one atom. Taking an arbitrary L signals from G_{j_1}, they span the same L-dimensional subspace as the atoms in Ω_{j_1}. Thus, gathering these L signals into a matrix $\mathbf{B}_{j_1} \in \mathbb{R}^{n\times L}$, there exists a vector $\mathbf{v}_1 \in \mathbb{R}^{L\times 1}$ such that $\mathbf{B}_{j_1}\mathbf{v}_1$ is parallel to the desired intersection atom. Similarly, taking L arbitrary members from G_{j_2} and forming the matrix $\mathbf{B}_{j_2} \in \mathbb{R}^{n\times L}$, there exists a vector $\mathbf{v}_2 \in \mathbb{R}^{L\times 1}$ such that $\mathbf{B}_{j_2}\mathbf{v}_2$ is parallel to the same intersection atom. Thus, we have the relationship

$$\mathbf{B}_{j_1}\mathbf{v}_1 = \mathbf{B}_{j_2}\mathbf{v}_2, \tag{15}$$

or posed differently, this relationship leads to the homogeneous linear system of equations

$$[\mathbf{B}_{j_1}, -\mathbf{B}_{j_2}]\mathbf{v} = 0, \tag{16}$$

where $\mathbf{v}$ is a vertical concatenation of $\mathbf{v}_1$ and $\mathbf{v}_2$. The constructed matrix has n rows and $2L$ columns, but we already know that its rank is $2L-1$ due to the intersection. A vector $\mathbf{v}$ in its null-space leads to the desired $\mathbf{v}_1$ and $\mathbf{v}_2$, and they can be obtained as the last right singular vector in an SVD operation [43]. Having found $\mathbf{v}_1$, the term $\mathbf{B}_{j_1}\mathbf{v}_1$ stands for the desired intersection atom, being parallel to a true atom found in $\mathbf{A}$ up to a scalar multiplication.

Repeating the above process for each pair with a single atom intersection, we obtain $J_1 \cdot k$ candidate atoms. Starting with the first, we sweep through this set and seek all others that are parallel to it, pruning them. This process proceeds for the second, third, and till we remain with only k atoms. These are the desired atoms, being the columns of the original dictionary $\mathbf{A}$.

Up until now we have presented a constructive algorithm for the extraction of the dictionary $\mathbf{A}$ that was used in the construction of $\mathbf{B}$. Indeed, in the described algorithm there are multiple possibilities of choosing the pairs in the second stage, as also choosing the L elements that construct the matrices $\mathbf{B}_{j_1}$ and $\mathbf{B}_{j_2}$ in the third stage. Nevertheless, all possible choices lead to the the same solution $\mathbf{A}$ up to simple transformations. Thus, the matrix $\mathbf{B}$, created as the product $\mathbf{AX}$, is factorized as desired.

Could there be a different feasible factorization? Let us assume that there exists such a second different factorization $\mathbf{B} = \widetilde{\mathbf{A}}\widetilde{\mathbf{X}}$. Executing the above algorithm on $\mathbf{B}$ must lead to the matrix $\widetilde{\mathbf{A}}$, due to the constructive method we have developed. On the other hand, this algorithm must also lead to $\mathbf{A}$ for the same reasons. Thus, we necessarily conclude that $\widetilde{\mathbf{A}}$ must be equivalent to $\mathbf{A}$, and therefore the factorization is unique. □

Let us now return and revisit the assumptions made earlier, and in particular the daunting number of signals required to guarantee the success of the algorithm we have described. While $\binom{k}{L}(L+1)$ examples indeed suffice to guarantee uniqueness, far less examples could be used in practice. We have seen that in the developed algorithm there are severe redundancies in building the atoms. How much lower could this number of example go and still lead to successful factorization? As an example that illustrates the possibilities, we could take exactly $L+1$ examples per each group, but consider only $2k$ such groups. If those groups are chosen smartly to divide into pairs that overlap on each of the k atoms, this would be sufficient for the success of the algorithm. Thus, $2k(L+1)$ signal examples could in principle be used. In fact, even such set could be found redundant, due to the ability to cross pairs differently and exploit other overlaps.

5. The K-SVD algorithm

The algorithm for extracting the underlying sparse factorization described in the previous sections is impractical, as it requires unreasonable computational effort (searching over all subsets of $L+1$ input signals). Hence, we now leave the theoretical discussion and present a different, more practical, algorithm. In this section we introduce the K-SVD algorithm for sparse matrix factorization. This algorithm aims to extract the dictionary matrix $\mathbf{A}$ and the sparse coefficient matrix $\mathbf{X}$. The K-SVD algorithm is flexible, and works in conjunction with any pursuit algorithm. It is simple, and designed to be a truly direct generalization of the K-Means clustering algorithm. As such, when forced to work with one atom per signal, and have a unit coefficient for this atom, it exactly reproduces the K-Means algorithm. Of course, it suffers from all the shortcomings of the K-Means, and in particular, it might be trapped into a local minimum solution.

We start our discussion with a short description of the vector quantization (VQ) problem and the K-Means algorithm. In VQ, a codebook $\mathbf{C}$ that includes k codewords is used to represent a wide family of signals $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^{N}$ $(N \gg k)$ by a nearest neighbor assignment. This leads to an efficient compression or description of those signals, as clusters in $\mathbb{R}^n$ surrounding the chosen codewords. The VQ problem can be formally described by

$$\min_{\mathbf{C},\mathbf{X}} \|\mathbf{B}-\mathbf{C}\mathbf{X}\|_{\mathrm{F}}^2 \quad \text{subject to } \{\|\mathbf{x}_i\|_0 = 1\}_{i=1}^N \ \wedge\ \mathbf{X} \in \{0,1\}^{k\times N}. \tag{17}$$

The *K-Means* algorithm [44] is an iterative method used for solving the above problem. At each iteration there are two stages—one for sparse coding that essentially evaluates $\mathbf{X}$ by mapping each signal to its closest atom in the current $\mathbf{C}$, and a second stage for updating the codebook, changing sequentially each column $\mathbf{c}_i$ in order to better represent the signals mapped to it. At each such stage, the penalty in (17) is minimized, squeezing the best out of the given conditions. As this error is bounded from below by zero, and the algorithm ensures a monotonic decrease of this value, convergence to at least a local minimum solution is guaranteed.

The sparse representation problem can be viewed as a generalization of VQ objective (17), in which we allow each input signal to be represented by a *linear combination* of codewords, which we now call atoms. As a consequence, the minimization described in Eq. (17) converts to

$$\min_{\mathbf{A},\mathbf{X}} \|\mathbf{B}-\mathbf{A}\mathbf{X}\|_{\mathrm{F}}^2 \quad \text{subject to } \{\|\mathbf{x}_i\|_0 \leqslant L\}_{i=1}^N. \tag{18}$$

In the K-SVD algorithm we solve (18) iteratively, using two stages, parallel to those in the K-Means. In the sparse coding stage, we compute the coefficients matrix $\mathbf{X}$, using any pursuit method, while allowing each coefficient vector to have no more than L non-zero elements. Referring to (18) and assuming that $\mathbf{A}$ is known, we should solve a set of N independent problems of the form

$$\left\{\min_{\mathbf{x}_i} \|\mathbf{b}_i-\mathbf{A}\mathbf{x}_i\|_{\mathrm{F}}^2 \quad \text{subject to } \|\mathbf{x}_i\|_0 \leqslant L\right\}_{i=1}^N. \tag{19}$$

Returning to our discussion from Section 2, pursuit algorithms such as the basis pursuit or matching pursuit could approximate a solution for these problems.

In the second stage we update each dictionary element sequentially, changing its content, along with the values of its coefficients, so as to better represent the signals that use this column. This sequential update of the columns and its relevant coefficients results in a Gauss–Seidel-like acceleration, since the subsequent columns to consider for updating are based on more relevant coefficients. The process of updating each column $\mathbf{a}_k$ has a straight forward solution based on the SVD [43]. Assume that the sub-matrix $\mathbf{B}_k$ contains in its columns the signal examples that use the kth atom $\mathbf{a}_k$. We start by computing the residual matrix

$$\mathbf{E}_k = \mathbf{B}_k - (\mathbf{A}\mathbf{X}_k - \mathbf{a}_k\mathbf{x}^k). \tag{20}$$

The sub-matrix $\mathbf{X}_k$ holds the representation coefficients for the examples in $\mathbf{B}_k$. The row vector $\mathbf{x}^k$ contains the coefficients that multiply the kth atom. We seek a new pair $\{\mathbf{a}_k, \mathbf{x}^k\}$ that will best approximate this residual, namely minimizing the error $\|\mathbf{E}_k - \mathbf{a}_k\mathbf{x}^k\|_{\mathrm{F}}$. This is a rank-1 approximation of $\mathbf{E}_k$ that can be found by the singular

value decomposition [43]. Note that in using SVD we necessarily have that (i) the columns of **A** are obtained already normalized; and (ii) the support of all representations either stays the same or get smaller by possible nulling of terms.

We called this algorithm "K-SVD" to parallel the name K-Means. While K-Means applies k mean calculations to evaluate the codebook, the K-SVD obtains the updated dictionary by k SVD operations, each producing one column. A more thorough discussion concerning variations of the K-SVD, implementation details and validation tests can be found in [40].

6. Applications to image processing

We performed several experiments that involve true image data, trying to show the practicality of the tools in hand using the proposed algorithm and the general sparse coding theme. We should emphasize that our tests here come only to demonstrate the concept of using such factorizations in sparse representations, and further work is required to fully deploy those ideas in actual applications. The example signals were gathered as a set of 11,000 examples of image block patches of size 8×8 pixels, taken from a database of face images (in various locations). These blocks constructs the signal matrix $\mathbf{B} \in R^{64\times 11,000}$, for which we would like to find a sparsely representing dictionary (i.e., a factorization $B = AX$ as in Eq. (18)). Working with real images data we preferred that all dictionary elements except one has a zero mean. Therefore, the first dictionary element was set to include a constant value in all its entries, and was not changed afterwards. This element takes part in all representations, and as a result, all other dictionary elements remain with zero mean during all iterations. We applied the K-SVD, training a dictionary of size 64×441 (the matrix **A**). The choice $k = 441$ came from our attempt to compare the outcome to the undecimated overcomplete Haar dictionary of the same size [20]. This dictionary has separable basis functions, having steps of various sizes and in all locations. The trained dictionary elements are shown in Fig. 1. The coefficients were computed using the OMP, where the maximal number of coefficients is $L = 10$. Note that better performance can be obtained by switching to BP or FOCUSS. We concentrated on OMP because of its simplicity and fast execution.

Filling in missing pixels

We chose one random full face image that contains ∼600 blocks (all of which were not used for training). On each block a ratio of the pixels (between 10% and 90%), in random locations, were discarded. In the recovery process, each of the corrupted blocks is treated independently, seeking a sparse representation to compose it with an ℓ^2 error of 5 gray-levels per pixel. In this decomposition process, obtained by the OMP, only the available pixels take part, thus using only the relevant rows from the dictionary **A** that correspond to these pixels. Normalization of this sub-dictionary columns is

necessary so as to fascilitate a fast OMP usage. Once the sparse representation for the block is found, multiplication by the full original dictionary gives an approximation of the missing pixels.

The mean reconstruction errors (for all blocks and all corruption rates) were computed, and are displayed in Fig. 2. Three test images and their reconstructions can be seen in Fig. 3, where the left image is the corrupted one, having some of its pixels missing, and the middle and right images present the reconstructed images by the learned and Haar dictionaries, respectively. As can be seen, high quality recovery is obtained, and with substantial advantage to the learned dictionary. Note that in using bigger blocks the performance would have been further improved.

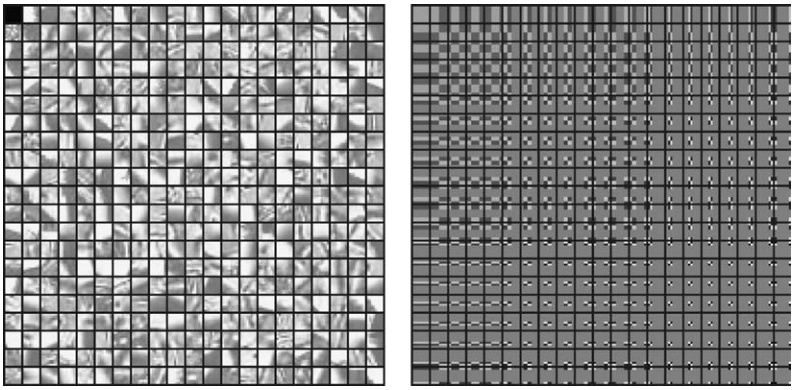

Fig. 1. K-SVD resulted dictionary (left) and the Haar dictionary (right).

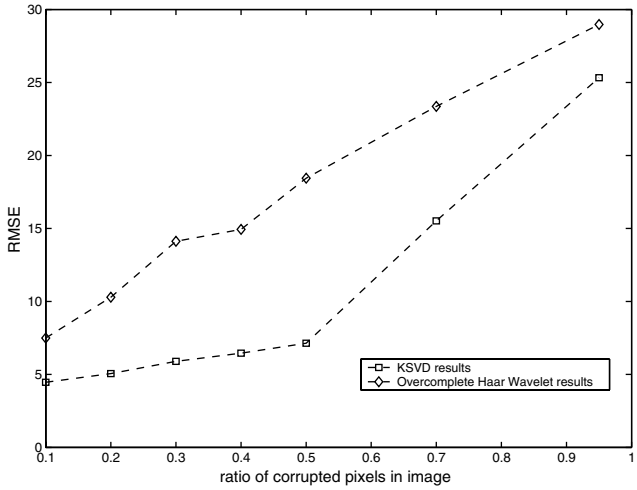

Fig. 2. Filling in - RMSE versus the relative number of missing pixels.

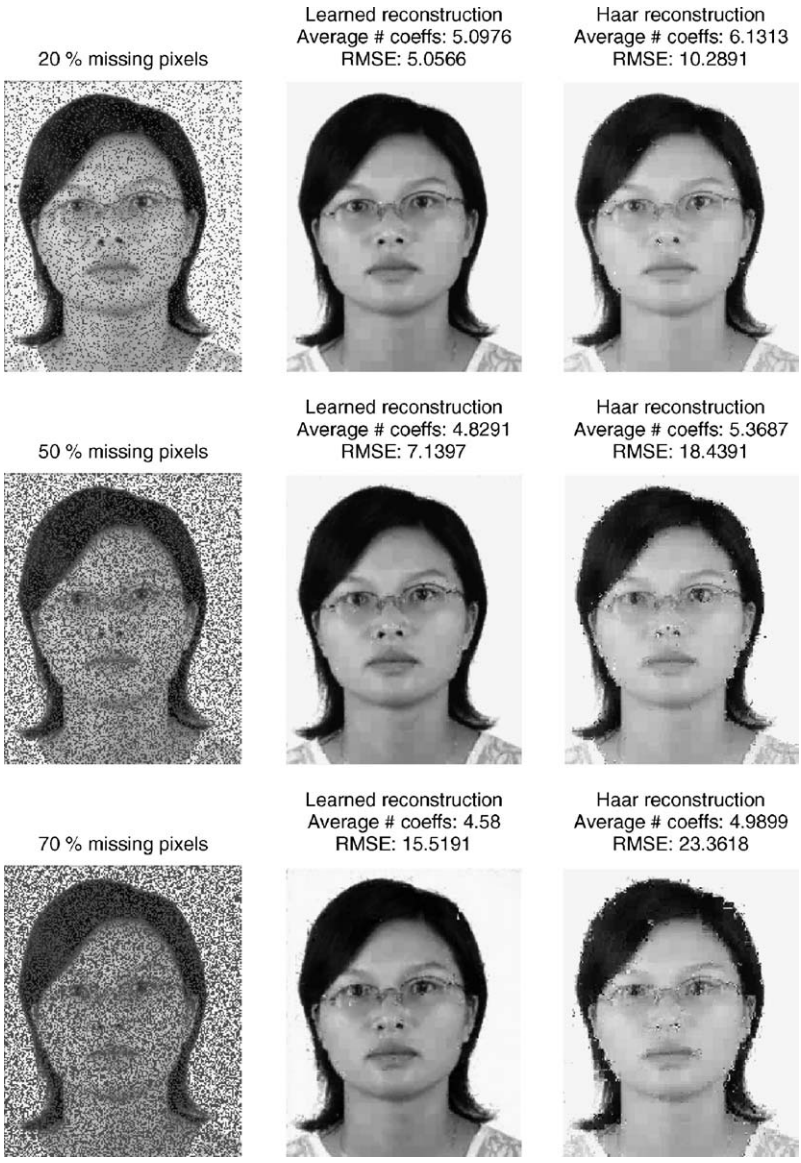

Fig. 3. Filling in Results. Black pixels in the images on the left indicate missing values.

Compression

A compression test was conducted, comparing between the learned dictionary, the overcomplete Haar dictionary, and the complete DCT dictionary, which is being used

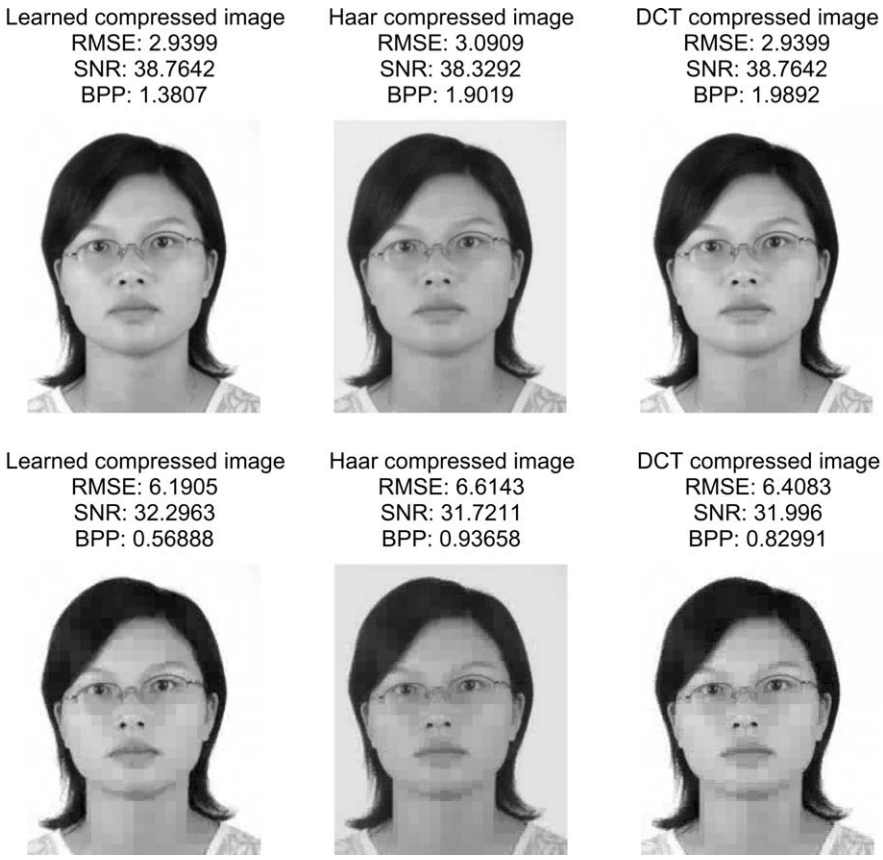

Fig. 4. Compression results.

by JPEG algorithm. Each 8×8 block was compressed independently until it holds a specified SNR value. The decomposition was done using OMP with an error bound (that depends on the desired SNR value). The compression was measured in bit per pixel (BPP), assuming 10 bits per each coefficient. In the overcomplete dictionaries, elements with arbitrary index were allowed, and therefore, the BPP value was set as $\mathrm{BPP} = C(10 + \log k)/64$ (where C is the number of required coefficients). In DCT, we used the leading coefficient in a zig-zag order (as done by JPEG), resulting $\mathrm{BPP} = C \cdot 10/64$.

Sample compressed images can be seen in Fig. 4, and a summary rate-distortion graph is presented in Fig. 5. We can see that the learned dictionary outperforms the other two alternatives in all tested compression rates.

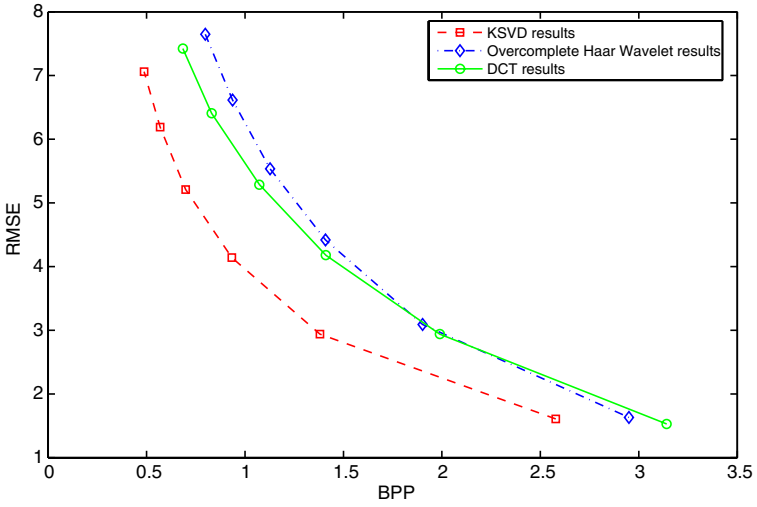

Fig. 5. Compression results in a graph.

7. Conclusions

In this paper we have studied the problem of dictionary design for sparse representations. We posed this task as a matrix factorization problem under sparsity constraint. We first proved the uniqueness of such a factorization. Doing so, we provided the theoretical background for constructing a practical algorithm for extracting this factorization. We then presented the K-SVD algorithm which aims to perform this task. This algorithm is a generalization of the K-means used frequently for clustering. We presented experiments that deploy the K-SVD and the ideas of sparsity and redundant representations to real images.

We strongly believe in the importance and the potential encompasses in the questions and solutions raised in this paper, and our plan is to proceed this work and explore further topics around these issues.

Acknowledgments

This research was partially supported by the Technion Fund for Promotion of Research—Charles Krown Research Fund, and by the Applied Materials donation.

References

[1] G. Davis, S. Mallat, M. Avellaneda, Adaptive greedy approximations, J. Constr. Approx. 13 (1997) 57–98.

[2] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, Int. J. Control 50 (5) (1989) 1873–1896.

[3] S. Mallat, Z. Zhang, Matching pursuits with time–frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[4] G. Davis, S. Mallat, Z. Zhang, Adaptive time–frequency decompositions, Optical-Engineering 33 (7) (1994) 2183–2191.
[5] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Conference Record of the Twenty Seventh Asilomar Conference on Signals, Systems and Computers, vol. 1, 1993.
[6] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 129–159.
[7] B.D. Rao, K. Kreutz-Delgado, Deriving algorithms for computing sparse solutions to linear inverse problems, in: Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, vol. 1, IEEE, 1998, pp. 955–959.
[8] D.L. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Inform. Theory 47 (7) (1999) 2845–2862.
[9] M. Elad, A.M. Bruckstein, A generalized uncertainty principle and sparse representation in pairs of bases, IEEE Trans. Inform. Theory 48 (9) (2002) 2558–2567.
[10] D.L. Donoho, M. Elad, Optimally sparse representation in general (non-orthogonal) dictionaries via l_1 minimization, PNAS 100 (5) (2003) 2197–2202.
[11] R. Gribonval, M. Nielsen, Sparse decompositions in unions of bases, IEEE Trans. Inform. Theory 49 (12) (2003) 3320–3325.
[12] J.J. Fuchs, On sparse representations in arbitrary redundant bases, IEEE Trans. Inform. Theory 50 (6) (2004) 1341–1344.
[13] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inform. Theory 50 (10) (2004) 2231–2242.
[14] D.L. Donoho, M. Elad, V. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, IEEE Trans. Inform. Theory, submitted for publication.
[15] J.A. Tropp, Just relax: convex programming methods for subset selection and sparse approximation, Technical Report 04-04, ICES Report, UT-Austin, 2004.
[16] D.L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution, Technical Report no. 2004-10, Department of Statistics, Stanford University, 2004.
[17] D.L. Donoho, For most large underdetermined systems of linear equations, the minimal ℓ^1-norm near-solution approximates the sparsest near-solution, Technical Report no. 2004-11, Department of Statistics, Stanford University, 2004.
[18] E. Candès, J. Romberg, Quantitative robust uncertainty principles and optimally sparse decompositions, Found. Comput. Math., to appear.
[19] M. Elad, M. Zibulevsky, A probabilistic study of the average performance of the basis pursuit, IEEE Trans. Inform. Theory, submitted for publication.
[20] D.S. Taubman, M.W. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, Norwell, MA, USA, 2001.
[21] D.L. Donoho, I.M. Johnstone, Ideal denoising in an orthonormal basis chosen from a library of bases, Comptes Rendus del'Academie des Sciences, Ser. A 319 (1994) 1317–1322.
[22] R. Coifman, D.L. Donoho, Translation invariant denoising, Wavelets Statist., Lecture Notes in Statistics 103 (1995) 120–150.
[23] E.P. Simoncelli, W.T. Freeman, E.H. Adelsom, D.J. Heeger, Shiftable multi-scale transforms, IEEE Trans. Inform. Theory 38 (2) (1992) 587–607.
[24] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Process. 11 (2002) 670–684.
[25] R. Gastaud, J.L. Starck, Dynamic range compression: a new method based on wavelet transform, in: Astronomical Data Analysis Software and Systems Conference, Strasbourg, 2003.

[26] J.L. Starck, M. Elad, D.L. Donoho, Image decomposition: separation of texture from piece-wise smooth content, in: SPIE Conference on Signal and Image Processing: Wavelet Applications in Signal and Image Processing X, SPIE's 48th Annual Meeting, 3–8 August 2003, San Diego, 2003.
[27] J.L. Starck, M. Elad, D.L. Donoho, Image decomposition via the combination of sparse representations and a variational approach, IEEE Trans. Image Process., accepted for publication.
[28] M. Elad, J.L. Starck, P. Querre, D.L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (mca), J. Appl. Comput. Harmon. Anal., submitted for publication.
[29] B.A. Olshausen, D.J. Field, Natural image statistics and efficient coding, Network: Comput. Neural Syst. 7 (2) (1996) 333–339.
[30] B.A. Olshausen, B.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1?, Vision Res. 37 (1997) 3311–3325.
[31] M.S. Lewicki, B.A. Olshausen, A probabilistic framework for the adaptation and comparison of image codes, J. Opt. Soc. Amer. A 16 (7) (1999) 1587–1601.
[32] M.S. Lewicki, T.J. Sejnowski, Learning overcomplete representations, Neural Comp. 12 (2000) 337–365.
[33] K. Engan, S.O. Aase, J.H. Husøy, Multi-frame compression: theory and design, EURASIP Signal Process. 80 (10) (2000) 2121–2140.
[34] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T. Lee, T.J. Sejnowski, Dictionary leaning algorithms for sparse representation, Neural Comput. 15 (2) (2003) 349–396.
[35] S. Lesage, R. Gribonval, F. Bimbot, L. Benaroya, Learning unions of orthonormal bases with thresholded singular value decomposition, in: IEEE Conf. on Acoustics, Speech and Signal Processing, 2005.
[36] T. Strohmer, R.W. Heath, Grassmannian frames with applications to coding and communication, Appl. Comput. Harmon. Anal. 14 (2004) 257–275.
[37] I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using focuss: a re-weighted norm minimization algorithm, IEEE Trans. Signal Process. 45 (3) (1997) 600–616.
[38] B.D. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection, IEEE Trans. Signal Process. 47 (1) (1999) 187–200.
[39] B.D. Rao, K. Engan, S.F. Cotter, J. Palmer, K. Kreutz-Delgado, Subset selection in noise based on diversity measure minimization, IEEE Trans. Signal Process. 51 (3) (2003) 760–770.
[40] M. Aharon, M. Elad, A.M. Bruckstein, K-svd: an algorithm for designining of overcomplete dictionaries for sparse representation, IEEE Signal Process., submitted for publication.
[41] D.L. Donoho, M. Elad, On the stability of the basis pursuit in the presence of noise, EURASIP Signal Process. J., submitted for publication.
[42] S. Sardy, A.G. Bruce, P. Tseng, Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries, J. Comput. Graph. Statist. 9 (2000) 361–379.
[43] G.H. Golub, C.F. Van-Loan, Matrix Computations, third ed., John Hopkins University Press, Baltimor, 1996.
[44] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.