Mixed Integer Programming for Sparse Coding: Application to Image Denoising

Yuan Liu, Stephane Canu, Paul Honeine, Su Ruan

Mixed Integer Programming for Sparse Coding: Application to Image Denoising

Yuan Liu, Stéphane Canu, Paul Honeine, Su Ruan

Abstract—**Dictionary learning for sparse representations is generally conducted in two alternating steps: sparse coding and dictionary updating. In this paper, a new approach to solve the sparse coding step is proposed. Because this step involves an ℓ_0-norm, most, if not all existing solutions only provide a local or approximate solution. Instead, a real ℓ_0 optimization is considered for the sparse coding problem providing a global solution. The proposed method reformulates the optimization problem as a Mixed-Integer Quadratic Program (MIQP), allowing then to obtain the global optimal solution by using an off-the-shelf optimization software. Because computing time is the main disadvantage of this approach, two techniques are proposed to improve its computational speed. One is to add suitable constraints and the other to use an appropriate initialization. The results obtained on an image denoising task demonstrate the feasibility of the MIQP approach for processing real images while achieving good performance compared to the most advanced methods.**

Index Terms—**Mixed-integer quadratic programming, sparse representation, sparse coding, dictionary learning, image denoising, K-SVD**

I. INTRODUCTION

Learning sparse representations to model data, signals and images have been widely investigated since it was introduced 20 years ago by Olshausen and Field in [1]. In recent years, learning sparse representations have been successfully applied to signal and image processing, as well as computer vision tasks, such as image denoising, image inpainting, object recognition, face recognition and classification and many classification tasks (see for instance [2] and included references).

A sparse representation describes a given signal by a linear decomposition of a few elements of a dictionary. Beyond predefined dictionaries, such as wavelets and many variants [3], [4], data-driven constructed dictionaries allow to have well-adapted and more natural representations for the signals at hand. Dictionary learning consists in jointly estimating the dictionary elements (the so-called atoms) and their coefficients (*i.e.*, contributions) in the linear decomposition.

Given a signal $\mathbf{y} \in \mathbb{R}^n$, a sparse representation of $\mathbf{y}$ takes the form $D\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^p$ is a sparse vector of coefficients and $D = [\mathbf{d}_1, \ldots, \mathbf{d}_i, \ldots, \mathbf{d}_p] \in \mathbb{R}^{n \times p}$ is the

Y. Liu, S. Canu, P. Honeine and S. Ruan are with the Normandie Univ, INSA Rouen, UNIROUEN, UNIHAVRE, LITIS, Avenue de l'Université, 76801 Saint-Étienne-du-Rouvray, France. Email: yuan.liu@insa-rouen.fr, stephane.canu@insa-rouen.fr, paul.honeine@univ-rouen.fr, su.ruan@univ-rouen.fr

learned dictionary of p atoms. Imposing the sparsity is naturally conducted with the ℓ_0-norm, by controlling the number of non-null components of the vector $\mathbf{x}$ with $\|\mathbf{x}\|_0$ [5]. In practice, the dictionary and the sparse vector are estimated with an alternating strategy, by optimizing with respect to one variable, D or $\mathbf{x}$, while the other variable is fixed. While the construction of the dictionary becomes convex, this is not the case of the estimation of $\mathbf{x}$ (also called sparse coding), which is non-convex and NP-hard due to the ℓ_0-norm.

Several algorithms have been developed to learn dictionaries for sparse representations, the most known being K-SVD [5]. The K-SVD algorithm is a two-stage generalization of the k-means algorithm. The first stage operates sparse coding by a coordinate descent algorithm, for example matching pursuit (MP) or orthogonal matching pursuit (OMP). The second stage is dictionary updating by a singular value decomposition (SVD) algorithm. K-SVD can be viewed within the framework of projection theory [6]. Due to its good performance on image reconstruction, many variants of K-SVD have been developed in order to address different tasks. In [7], K-SVD is extended by a global image prior that leads to state-of-the-art denoising performances. Discriminative K-SVD algorithms are derived in [8] by investigating the classification accuracy. Other variants to improve the discriminative power of sparse representations include combining K-SVD with a kernel algorithm to deal with non-linear problems [9] and considering a graph-based regularization to account for the relationship among the atoms of dictionary [10].

Although K-SVD has earned a great success, the learned dictionary may be correlated and the good denoising performance can lead only if the noise information is known. K-SVD's Achilles heel is undoubtedly its sparse coding scheme, using either an approximate solution such as basis pursuit (BP), or a greedy solution such as MP or OMP. In the past years, research has been conducted to develop more relevant optimization techniques. Beyond greedy algorithms, the most promising working direction is proximal methods where the global optimal solution can be reached after a few number of iterations [11], [12]. In practice, proximal methods demonstrate a higher convergence speed than greedy algorithms with low computational complexity. Iterative thresholding algorithms for sparse recovery are special instances of the proximal algorithms [13]. Besides, some researchers update the dictionary learning problem using an efficient sum of outer products dictionary learning (SOUPDIL) [14], by representing $D[\mathbf{x}_1, \ldots, \mathbf{x}_\ell]$ as the sum of outer products $\sum_{i=1}^{p} \mathbf{d}_i \mathbf{c}_i^T$, with $[\mathbf{c}_1, \ldots, \mathbf{c}_p] = [\mathbf{x}_1, \ldots \mathbf{x}_\ell]^T$.

In this paper, following [15] and [16], the sparse coding

in its exact ℓ_0-norm formulation is recast as a mixed-integer quadratic programming (MIQP), namely a mixed-integer programming (MIP) with a quadratic objective function. MIP aims at solving optimization problems involving both integer and continuous variables. Even if the use of MIP for pattern recognition is not new [17], only very recently it has been investigated with success to obtain the sparse approximation of a signal [16], to generate fiducial marker [18], to perform multiple face tracking [19] and vehicle detection [20]. Moreover, MIP is of a high tolerance to noise, and even compared with the recently proposed SOUPDIL algorithm [14]. However, unlike specific formulations such as SOUPDIL, the proposed MIP formulation exhibits genericity and flexibility, for instance by facilitating the integration of additional constraints. Because of its computation complexity, the preliminary studies conducted in [16] were restricted to tiny toy data (120-sample synthesized signals). To the best of our knowledge, MIP has never been used to address problems in image processing, neither applied on real datasets.

We demonstrate in this paper that dictionary learning with K-SVD for image processing can be naturally achieved with mixed-integer programming, instead of the coordinate descent algorithm conventionally used in K-SVD. To this end, we recast the sparse coding problem as a mixed-integer programming with a quadratic objective function and linear constraints. We investigate recent theoretical progress in linear program and novel improvements of efficient implementation (see for instance [21]). To provide an efficient resolution of the resulting MIQP, we propose two techniques to increase the convergence speed by reducing the searching time and decreasing the boundary as well: include appropriate constraints and initialize with the proximal method. These improvements allow to reduce the computational complexity about ten to fifty times, thus confirming the feasibility of applying the algorithm to image processing. The relevance of these developments is demonstrated on well-known benchmark images frequently used in image processing, such as Barbara, Cameraman, Elaine, Lena and Man. Conducted experiments on image denoising show the tolerance to noise of the proposed MIQP algorithm, outperforming algorithms based on OMP and proximal methods.

This paper makes three main contributions:

- It demonstrates that the resolution of the sparse coding with K-SVD for dictionary learning can be done efficiently using the exact optimization method MIQP.
- It explores two techniques to speed-up the convergence of MIP, making it feasible to address image processing tasks.
- It shows that MIQP realizes image denoising on real benchmark images without any prior knowledge. Furthermore, it obtains the best denoising performance comparing with OMP and proximal methods.

The rest of the paper is organized as follows. The sparse representation problem and the classical algorithms are presented in Section II. The proposed MIQP-based dictionary learning algorithm and the theoretical analysis are described in Section III. In Section IV, experimental results on image denoising

show that the proposed method can train the dictionary with high tolerance to noise. The last section concludes this paper.

II. SPARSE REPRESENTATION OF SIGNALS

This section states several basic interprets of the sparse representation problems. To address those optimization problems, several popular algorithms for sparse coding and dictionary updating are presented.

A. Problem Statement

Considering a matrix $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_i, \ldots, \mathbf{y}_\ell] \in \mathbb{R}^{n \times \ell}$ of ℓ signals of dimension n, a sparse representation of Y consists in finding a matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_\ell] \in \mathbb{R}^{p \times \ell}$ of decomposition coefficients, which is sparse over a learned dictionary $D = [\mathbf{d}_1, \ldots, \mathbf{d}_p] \in \mathbb{R}^{n \times p}$. The columns of the latter, *i.e.*, $\mathbf{d}_j$ for $j = 1, \ldots, p$, are called atoms.

Sparse representations have been considered with success in signal and image processing. When working on a given image (or a set of images), it is fragmented into (often overlapping) patches, where each patch is unfolded to define a signal $\mathbf{y}_i$. Therefore, ℓ is the number of patches and n is the number of pixels for each patch, such as $n = 64$ when dealing with 8×8 overlapping patches.

Obviously, the set of signals is typically larger than its dimension, namely $\ell \gg n$. And in general, D is an overcomplete dictionary, that is to say $n < p$, while the situation $n > p$ is allowed for some discrimination tasks [22]. To prevent the ℓ_2-norm of dictionary's atoms from being arbitrarily large which leads to arbitrarily small decomposition coefficient in X, the dictionary D should be restricted in the constraint $\mathcal{C} = \{D \in \mathbb{R}^{n \times p} \text{ subject to } \mathbf{d}_j^T \mathbf{d}_j \leq 1, \ \forall j = 1, \ldots, p\}$. A pre-processing of the data, like centering, contrast normalization and whitening is often considered to impose some properties, such as transformation invariance, illumination invariance or some confounding effect removal [2]. Beyond these considerations, the sparse representation can be obtained by solving the following optimization problem:

$$\min_{D \in \mathcal{C}, \mathbf{x}_i \in \mathbb{R}^p} \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\frac{1}{2} \|\mathbf{y}_i - D\mathbf{x}_i\|_2^2 + \lambda \Omega(\mathbf{x}_i) \right). \quad (1)$$

The first term $\frac{1}{2}\|\mathbf{y}_i - D\mathbf{x}_i\|_2^2$ is the reconstruction error with $\| \ . \ \|_2$ being the Euclidean norm. The second one includes the regularization term $\Omega(\mathbf{x}_i)$ to enforce sparsity. The regularization parameter $\lambda > 0$ controls the trade-off between data fitting and sparsity of X. For the sake of clarity of this paper, the reconstruction error is measured with the square loss; generalization to other loss functions such as the logistic or hinge losses is straightforward [23]. Moreover, the work given in this paper can be extended to other tasks, such as classification where a discriminative term is introduced into the objective function to increase the discriminative power of the learned dictionary, *e.g.* a Fisher's criterion [2]. Generally, the regularization function Ω is associated to a norm that promotes sparsity and its formulation depends on the task at hand [12], [5]. A natural definition of Ω to promote sparsity is the ℓ_0 quasi-norm, *i.e.*, $\Omega(\mathbf{x}) = \|\mathbf{x}\|_0$, which refers to the number of non-zeros of $\mathbf{x}$.

The problem of estimating simultaneously X and D is non-convex and belongs to NP-hard problems. It is often solved via an alternating strategy: 1) fixing D and finding sparse coefficients X, the procedure is called sparse coding; 2) fixing X and search the solution of D, this is the procedure of dictionary updating. While the latter yields a convex optimization problem, the sparse coding is more difficult due to the sparsity constraint. Several popular algorithms for sparse coding and dictionary updating are described in the following, as well as the most known combinations of these algorithms.

B. Sparse coding

As introduced above, sparse coding consists of finding the sparse decomposition coefficients with a fixed dictionary, a procedure that can be easily parallelized by considering separately each signal. There exist several formulations to address this problem. These formulations are not equivalent in general due to the nonconvexity of the problem at hand; See [24] and references therein for more details. We consider in the following two well-known formulations, the sparsity-constrained and the error-constrained formulations.

The sparsity-constrained formulation is defined as follows. Since the dictionary D is fixed, the sparse coefficients $\mathbf{x}$ for each signal $\mathbf{y}$ can be obtained by solving the following problem:

$$\min_{\mathbf{x} \in \mathbb{R}^p} \tfrac{1}{2}\|\mathbf{y} - D\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq T. \tag{2}$$

This formulation is practical for solving the problem with prior knowledge of the sparsity level T of the signals (for more details, see [2], [16] and included references). At the same time, ℓ_1-norm or ℓ_2-norm is also sometimes used and it shows better performance in some application [22].

The error-constrained formulation of the sparse coding problem is often used for image denoising tasks:

$$\min_{\mathbf{x} \in \mathbb{R}^p} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \tfrac{1}{2}\|\mathbf{y} - D\mathbf{x}\|_2^2 \leq \varepsilon, \tag{3}$$

where ε can be a function of the noise level, for example, the standard deviation for the Gaussian white noise. For this reason, this formulation (3) is often considered in image denoising problems, since it allows to incorporate the (estimated) noise level [7].

Recently, the SOUPDIL algorithm [14] was proposed by varying the sparse representation problem to:

$$\min_{\mathbf{a}_i \in \mathbb{R}^p} \tfrac{1}{2}\|Y - \sum_{i=1}^p \mathbf{d}_i \mathbf{a}_i^T\|_F^2 + \lambda \sum_{i=1}^\ell \|\mathbf{a}_i\|_0 \\ \text{subject to} \quad \|\mathbf{a}_i\|_\infty \leq L \tag{4}$$

where $A = [\mathbf{a}_1, \ldots, \mathbf{a}_i, \ldots, \mathbf{a}_p] \in \mathbb{R}^{l \times p}$ and $A = X^T$. The parameter $L > 0$ avoids the non-coercive objective which is defined in [14] as a function of the Y, namely $L = \|Y\|_F$.

Unlike the problem described in (2), the problem (4) offer a total sparsity level of the signals but the variable sparsity across signals are allowed. This algorithm using block coordinate descent approach is proved to be efficient and get the promising performance.

Providing an exact solution for the ℓ_0-norm optimization problem is intractable in general. For this reason, approximate

optimization algorithms have been developed in the literature [25], and can be roughly grouped in two major classes: coordinate descent and gradient descent.

- Coordinate descent algorithms
  The two well-known coordinate descent algorithms are matching pursuit and orthogonal matching pursuit (OMP) [2]. Based on the projection theory, these recursive algorithms start with a null vector of coefficients. In each iteration, MP updates a single component, namely the one whose atom is the most correlated to the residual. OMP is widely used due to its efficacy [5], [26].
- Gradient descent algorithms
  Gradient descent algorithm [27] is another popular approximation method. It searches for the optimal solution by a descent in the gradient direction at each step. The Proximal gradient method is a more general algorithm, which is viewed as the tool for solving non-smooth or constrained optimization problems (*e.g.* sparse reconstruction problems). The proximal method has been widely used in image processing due to its convergence rate and ability of dealing with non-convex problems [28]. See Section III-C for more details.

Other optimization techniques have been introduced to solve the problem. For example, the method introduced in [29] extends the simplex method to the proposed the homotopy algorithm to solve a LASSO problem. Based on the piecewise linear property, a special regularization path is defined to lead to the optimal solution. Other approaches include probabilistic graphic models and the Bayesian framework to learn a dictionary with a series of sampling process. Recently in [30], the Bayesian non-parameter framework and Indian Buffet Process are used to learn the dictionary without setting the size of dictionary or noise level. As shown by the authors, its denoising performance is comparable with the SVD using the OMP algorithm.

C. Dictionary updating

Dictionary updating consists in estimating the dictionary D while the sparse coefficients X are fixed. The optimization problem (1) boils down to

$$\min_{D \in \mathcal{C}} \tfrac{1}{2}\|Y - DX\|_F^2, \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The use of a gradient descent algorithm becomes intractable when dealing with large-scale datasets, such as in image and video processing. To overcome this difficulty, two approaches have been largely investigated for dictionary updating: the stochastic gradient descent and the singular value decomposition methods.

- Stochastic gradient descent (SGD) algorithm
  Instead of dealing with all the samples at each iteration, SGD operates a gradient descent by estimating the gradient on the basis of a single randomly picked sample each time [25]. More precisely, let $\mathbf{y}_i$ be the selected sample at a given iteration, and $\mathbf{x}_i$ the corresponding sparse coding

vector. SGD updates the dictionary using the following rule

$$D_{k+1} = D_k - \eta(D_k \mathbf{x}_i - \mathbf{y}_i)\mathbf{x}_i^T, \qquad (6)$$

for a given step size parameter $\eta > 0$. In addition, the resulting dictionary matrix is projected onto $\mathcal{C}$ to fulfill the constraints. Some variants of this method include the use of a subset of samples at each iteration, namely the so-called mini-batch strategy [31].

- Singular value decomposition (SVD) algorithm
  When dealing with the update of a subset of atoms, one can solve the problem using SVD from linear algebra. Of particular interest is updating a single atom at each time, while keeping all the other atoms unchanged, as examined next. Let $\mathbf{d}_j$ be this atom, then the optimization problem (5) can be rewritten as

$$\min_{\mathbf{d}_j} \tfrac{1}{2}\|\big(Y - \sum_{\substack{i=1 \\ i \neq j}}^{p} \mathbf{d}_i \mathbf{x}^i\big) - \mathbf{d}_j \mathbf{x}^j\|_F^2,$$

where $\mathbf{x}^j$ is the j-th row of the matrix X. This is a simple rank-one optimization problem, since the solution is obtained by approximating the matrix between parentheses with a rank-one representation. This can be done efficiently, as given in [14, Proposition 3].

In this paper, we introduce a new algorithm for sparse coding with MIQP, in conjunction with SVD for dictionary updating. Next section presents the proposed MIQP algorithm.

III. Optimization Algorithm

As aforementioned, sparse coding addresses intrinsically a bi-objective optimization problem, where both sparsity and reconstruction error need to be optimized. So far, sparse coding has been tackled using approximate algorithms, such as a number of greedy algorithms and descent-based iterative hard thresholding. However, when put aside the computational complexity and memory usage, approximate algorithms fail to obtain the exact solution and are often very sensitive to additive noise.

In the following, we cast the sparse coding with MIQP in order to address the exact ℓ_0 optimization problem. Besides, with the development of the linear programming (LP) techniques and the improvements of the hardware's computational ability, the implementation speed is greatly improved [32]. Thus, applying MIQP to do sparse coding in the field of image processing becomes feasible.

A. MIQP

In the following, the sparse coding problem is addressed in its original formulation (2). This constrained optimization problem can be simplified, with all the entries of the sparse vector $\mathbf{x}$ indicated by a binary variable $\mathbf{z} \in \{0, 1\}^p$ [16], which can be explained by the logical relation:

$$\begin{cases} z_i = 0, & \text{if} \quad x_i = 0 \\ z_i = 1, & \text{if} \quad x_i \neq 0 \end{cases},$$

where z_i and x_i indicate the i-th entry of the vectors $\mathbf{z}$ and $\mathbf{x}$, $i = 1, \ldots, p$. Since such logical relation cannot be easily integrated into the objective function, we recast the sparsity condition into a linear inequality by introducing a sufficient big value $M > 0$ which should ensure that $\|\hat{\mathbf{x}}\|_\infty < M$ for any desirable solution $\hat{\mathbf{x}}$, where $\|\cdot\|_\infty$ is the maximum norm. A too big M will result in increased feasible region which will make the problem less computational efficient. An appropriate value of M improves the performance. The method to provide a lower M to obtain tight bounds will be discussed in the following.

Now the indicative function of $\mathbf{z}$ is ensured by satisfying the constraints:

$$-z_i M < x_i < z_i M, \quad \forall i \in \{1, \ldots, p\}. \qquad (7)$$

Then, the sparsity constraint $\|\mathbf{x}\|_0 \leq T$ in formulation (2) can be depicted by $\mathbf{z}$ as:

$$\sum_{i=1}^{p} z_i \leq T. \qquad (8)$$

As a consequence, the ℓ_0-based sparse coding problem (2) can have a 'big-M' reformulation, that is, for a given M large enough:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p, \mathbf{z} \in \{0,1\}^p} \quad & \tfrac{1}{2}\|\mathbf{y} - D\mathbf{x}\|_2^2 \\ \text{subject to} \quad & -\mathbf{z}M < \mathbf{x} < \mathbf{z}M \\ & \mathbf{1}_p^T \mathbf{z} \leq T, \end{aligned} \qquad (9)$$

where $\mathbf{1}_p$ is the column vector of size p with all elements equal to one. In this formulation, the optimization variables $\mathbf{x}$ and $\mathbf{z}$ are respectively continuous and integer. The problem is a mixed-integer program (MIP).

MIP refers to the optimization problems involving both integer and continuous variables. And according to the objective function (linear or quadratic) and constraints (linear, quadratic, equation or inequality), MIP can be further divided into different optimization problems, the most known are the Mixed-Integer Linear Program (MILP), the Mixed-Integer Quadratic Program (MIQP) and the Mixed-Integer Quadratically Constrained (linear) Program (MIQCP). Back to our optimization problem defined in (9) with continuous and integer optimization variables, the objective function is quadratic and all the constraints are linear. Hence, sparse coding can be interpreted as a MIQP. In the following, we will write the optimization problem in a standard form in order to use off-the-shelf solvers.

The standard formulation of MIQP is:

$$\begin{aligned} \min_{\mathbf{v}} \quad & \tfrac{1}{2}\mathbf{v}^T Q \mathbf{v} + \mathbf{c}^T \mathbf{v} \\ \text{subject to} \quad & A_{in}\mathbf{v} \leq \mathbf{b}_{in} \\ & A_{eq}\mathbf{v} = \mathbf{b}_{eq} \\ & \mathbf{l}_b \leq \mathbf{v} \leq \mathbf{u}_b \\ & v_j \in \mathcal{Z} \quad \forall j \in \mathcal{I}, \end{aligned} \qquad (10)$$

where $\mathbf{v}$ is the vector of optimization variables, with the set $\mathcal{I}$ indicating its integer components, Q is a symmetric matrix defining the quadratic objective function and $\mathbf{c}$ is its linear part, A_{in}, $\mathbf{b}_{in}$, A_{eq}, $\mathbf{b}_{eq}$, jointly define the constraints, $\mathbf{l}_b$ and $\mathbf{u}_b$ are the lower and upper bounds of the optimization

variables $\mathbf{v}$. The constraints determined by the two boundaries are fundamental in the complexity sense; without bounded constraints, the problem becomes undecidable [33]. In contrast, by raising the lower bounds and reducing the upper bounds, the computation complexity can be easily decreased [34].

We can reformulate our problem as a standard formulation of MIQP by combining the vectors $\mathbf{x}$ and $\mathbf{z}$, that is, let

$$\mathbf{v} = (\mathbf{x}^T, \mathbf{z}^T)^T,$$

then

$$\begin{aligned}
\min_{\mathbf{v}} \quad & \tfrac{1}{2}\mathbf{v}^T Q \mathbf{v} + \mathbf{c}^T \mathbf{v} \\
\text{subject to} \quad & A_{in}\mathbf{v} \leq \mathbf{b}_{in} \\
& v_j \in \{0,1\} \quad \forall j \in \mathcal{I},
\end{aligned} \tag{11}$$

where Q is a matrix of size $2p \times 2p$ made up of four sub-matrices

$$Q = \begin{pmatrix} D^T D & \mathbf{0}_{p,p} \\ \mathbf{0}_{p,p} & \mathbf{0}_{p,p} \end{pmatrix},$$

with $\mathbf{0}_{p,q}$ is the zero matrix of size $p \times q$, $\mathbf{c}$ is a column vector of size $2p$ with

$$\mathbf{c} = \begin{pmatrix} -D^T \mathbf{y} \\ \mathbf{0}_{p,1} \end{pmatrix}.$$

The $(2p+1) \times 2p$ matrix

$$A_{in} = \begin{pmatrix} -I_p & -MI_p \\ I_p & MI_p \\ \mathbf{0}_p^T & \mathbf{1}_p^T \end{pmatrix}$$

with I_p the identity matrix of size $p \times p$, and the $(2p+1) \times 1$ column vector $\mathbf{b}_{in} = (\mathbf{0}_{2p}^T, T)^T$, are both obtained according to the inequality of formulation (9). Finally, the set $\mathcal{I}$ in (10) indicates the integer components in the MIQP, namely

$$\mathcal{I} = \{p+1, p+2, \ldots, 2p\}.$$

In practice, the variable's type is indicated as continuous or binary in the input for the solver at hand.

To solve this MIQP problem, various optimization software packages can be explored, for example CPLEX developed by IBM integrates the latest MIP solvers to solve larger MILP problems, and Gurobi Optimizer recently developed can have an equivalent performance to CPLEX while the latest release gets some improvements [32]. The developed tools make it possible to apply MIQP into image processing, but by considering its computational complexity, some effort can be done to improve it as described next.

B. Additive constraints

The developments of the MIQP solvers have been following the progress in LP theory. The advanced-start capabilities of simplex algorithms in the branch-and-bound [35] (or now more correctly, branch-and-cut [34]) search tree are well exploited by MIQP solvers. No matter which optimization technique is used, the search process remains the main time consumption factor. The searching time heavily relies on the feasible region determined by the constraints. Hence, the effort on getting a good formulation of the constraints do help to accelerate the resolution of the optimization problem.

Hoffman and Ralphs have proven in [36] that, if a feasible solution is obtained by a relaxation, then it must also be optimal solution to the original problem. Especially, in the ideal case, if the convex envelope is found, a mixed integer programming will be transformed to the classical linear programming. However, it is an NP-hard problem to find constraints defining the convex envelop. The viable strategy is to create a convex envelop of the continuous variables

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^p \mid \mathbf{z} \in \{0,1\}^p, \sum_{j=1}^{p} z_j \leq T, |\mathbf{x}_j| \leq z_j T, \right\},$$

by adding the constraint about ℓ_1-norm and ℓ_∞-norm of $\mathbf{x}$:

$$\begin{cases} \sum_{i=1}^{p} |x_i| & < & TM \\ |x_i| & < & M \qquad \forall i = 1, \ldots, p. \end{cases}$$

However, the absolute value is difficult to be formulated as linear programs. To overcome this difficulty, we replace each unrestricted variable x_i, for $i = 1, \ldots, p$, with the difference of two restricted variables,

$$\begin{cases} x_i = x_i^+ - x_i^- \\ x_i^+, x_i^- \geqslant 0, \end{cases}$$

namely in matrix form

$$\begin{cases} \mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- \\ \mathbf{x}^+, \mathbf{x}^- \geqslant 0. \end{cases}$$

Then the absolute value of x_i in the above constraints can be represented in the linear program as:

$$|x_i| = x_i^+ + x_i^- \qquad \forall i = 1, \ldots, p.$$

Thus, the constraints for MIQP can be summarized as:

$$\begin{cases} \sum_{i=1}^{p} x_i^+ + x_i^- & < TM \\ -\mathbf{z}M < \mathbf{x}^+ - \mathbf{x}^- & < \mathbf{z}M \\ 0 \leq x_i^+, x_i^- & < M \\ \mathbf{1}_p^T \mathbf{z} & \leq T. \end{cases} \tag{12}$$

With the new constraints, MIQP can be reformulated as the standard formulation by introducing as updated optimization variable $\mathbf{v} = (\mathbf{x}^{+T}, \mathbf{x}^{-T}, \mathbf{z}^T)^T$. Accordingly, the model components Q, $\mathbf{c}$, A_{in}, $\mathbf{b}_{in}$, $\mathbf{l}_b$ and $\mathbf{u}_b$ are updated as follows: The matrix Q becomes the $3p \times 3p$ matrix

$$Q = \begin{pmatrix} D^T D & -D^T D & \mathbf{0}_{p,p} \\ -D^T D & D^T D & \mathbf{0}_{p,p} \\ \mathbf{0}_{p,p} & \mathbf{0}_{p,p} & \mathbf{0}_{p,p} \end{pmatrix},$$

the vector $\mathbf{c}$ changes to the vector of size $3p$

$$\mathbf{c} = \begin{pmatrix} -D^T y \\ D^T y \\ \mathbf{0}_{p,1} \end{pmatrix},$$

the linear constraint matrix $A_{in} \in \mathbb{R}^{(2p+2) \times 3p}$ is now

$$A_{in} = \begin{pmatrix} \mathbf{1}_p 6T & \mathbf{1}_p^T & \mathbf{0}_p^T \\ \mathbf{0}_p^T & \mathbf{0}_p^T & \mathbf{1}_p^T \\ -I_p & I_p & -Mi_p \\ I_p & -I_p & -MI_p \end{pmatrix},$$

the right side of the inequality constraint becomes

$$\mathbf{b}_{in} = \begin{pmatrix} TM \\ T \\ \mathbf{0}_{2p}^T \end{pmatrix},$$

the two bounds $(\mathbf{u}_{in}, \mathbf{l}_b)$ of the new variables $\mathbf{v}$ are now defined respectively as $\mathbf{u}_{in} = (M\mathbf{1}_{2p}^T, \mathbf{1}^T)^T$ and $\mathbf{l}_b = \mathbf{0}_{3p}$, and

$$\mathcal{I} = \{2p + 1, 2p + 2, \ldots, 3p\}.$$

With the new formulation, the problem can be solved more efficiently.

C. Initialization by the proximal method

The MIQP solver is based on the search tree theory [34]. The MIQP problem, represented by the root of the tree, is partitioned into subproblems. And the feasible region is also divided into subregions. The objective value of any feasible solution to a subproblem provides an upper bound on the global optimal value. The optimal solution is produced when the global lower bound and global upper bound are equal. Usually, a global bound is needed to make the algorithm more efficient. Hence, a good initialization or tight bounds can both help to improve the performance. In the following, the proximal method will be applied to give a good initialization and an optimized value for M that forms the global bound of the problem.

The proximal method is based on the first order approximation method. It produces a reasonable approximate solution by minimization a succession of upper bound of the objective function. It is a powerful tool to tackle non-smooth, constrained, large-scale, and distributed optimization problems. The proximal operator is expressed as

$$\text{prox}_h(\mathbf{u}) = \arg\min_{\mathbf{x}} \left(h(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 \right),$$

where h defines a proper and lower semi-continuous function, and $t > 0$ is a step size parameter. See [28] for more details.

For our problem, let $H(\mathbf{x})$ denote the quadratic objective function in the optimization problem (10), and $h(\mathbf{x})$ the function that makes sure that the feasible region is in the space $\mathcal{S}$ of T-sparsity, that is

$$h(\mathbf{x}) = \begin{cases} 0 & \text{if } \|\mathbf{x}\|_0 \leq T \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

The proximal operator boils down to the projection onto the sparse space $\mathcal{S}$:

$$P_{\mathcal{S}}(\mathbf{u}) := \arg\min_{\mathbf{x} \in \mathcal{S}}(\|\mathbf{x} - \mathbf{u}\|^2).$$

The solution of this problem can be easily obtained by keeping T biggest absolute value components of $\mathbf{u}$ and setting the rest to zeros:

$$P_{\mathcal{S}}(\mathbf{u}) = \begin{cases} u_j & \text{if } j \in \{(1), \ldots, (T)\} \\ 0 & \text{otherwise} \end{cases}$$

where j is the index of the sequences that $|u_{(1)}| \geqslant |u_{(2)}| \geqslant \cdots \geqslant |u_{(p)}|$. By applying a proximal algorithm, the sparse

---

Algorithm 1 Dictionary learning algorithm via MIQP.

Require: Signals for training Y, target sparsity T, step size t for updating approximate X by proximal method, coefficient α for optimizing M, number of iteration for dictionary learning N_d and for proximal method N_p

1: Initialize the dictionary D and the decomposition coefficients matrix X^0;
2: **for** $n_d = 1, \ldots, N_d$ **do**
3:    **Initialize X by proximal method:**
4:    **for** $n_p = 1, \ldots, N_p$ **do**
5:
$$X^{k+1} = P_{\mathcal{S}}\left(X^k - t(D^T D X^k - D^T Y) \right)$$

6:    **end for**
7:    $X = X^{N_p}$
8:    $M = \alpha \max_{i=1,\ldots,N} \|\mathbf{x}_i^{N_p}\|_\infty$,
9:    **Optimize X by MIQP:**
10:   **for** $n = 1, \ldots, N$ **do**
11:      $\mathbf{x}_n^+ = \max(0, \mathbf{x}_n)$
12:      $\mathbf{x}_n^- = \max(0, -\mathbf{x}_n)$
13:      $\mathbf{z}_n = \text{abs}(\text{sign}(\mathbf{x}_n))$
14:      $\mathbf{v} = (\mathbf{x}_n^{+T}, \mathbf{x}_n^{-T}, \mathbf{z}_n^T)^T$
15:      Solving MIQP problem

$$\min_{\mathbf{v}} \quad \frac{1}{2}\mathbf{v}^T Q \mathbf{v} + \mathbf{c}^T \mathbf{v}$$
$$\text{subject to} \quad A_{in}\mathbf{v} \leq \mathbf{b}_{in},$$
$$\mathbf{l}_b \leq \mathbf{v} \leq \mathbf{u}_b,$$
$$v_j \in \{0, 1\}, \quad \forall j \in \mathcal{I}.$$

$$\mathbf{x}_n = \mathbf{v}[1:p] - \mathbf{v}[p+1:2p]$$
16:   **end for**
17:   Update D with the SVD method
18: **end for**
19: **return** The dictionary D and the decomposition coefficients matrix X.

---

representation problem can be solved through a serial update process [12]:

$$\mathbf{x}^{k+1} \in P_{\mathcal{S}}\left(\mathbf{x}^k - t\nabla H(\mathbf{x}^k) \right), \tag{14}$$

where $\nabla H(\mathbf{x}) = D^T D \mathbf{x} - D^T \mathbf{y}$. After a finite number of iterations n_{iter}, the $\mathbf{x}^{n_{iter}}$ will be much approximate to the optimal solution of the MIQP problem. By considering the definition of 'big-M', the constraints in the problem (12) related to M can be well determined by an approximation of the optimal solution. A simple method to determine an appropriate value for M can be:

$$M = \alpha\|\mathbf{x}^{n_{iter}}\|_\infty. \tag{15}$$

The much tighter bound defined by M and an approximate initialization allow to gain a factor of ten in the required computing time for solving such problems (this figure is obtained from extensive preliminary experiments conducted on toy data).

The optimized dictionary learning algorithm based on MIQP is outlined as given in Algorithm 1.

IV. EXPERIMENTS AND RESULTS

Sparse representation on image denoising has been addressed by methods such as K-SVD and BM3D [37], and it has

been recently surpassed by the methods of convolutional neural networks [38]. However, the experiment on image denoising still provides a good assessment on sparse representation algorithms [39]. In this section, we design two experimental settings. The first one is conducted on synthetic data in order to show the advantage of sparse coding with MIQP, considering both accuracy and speed. The second one evaluates the dictionary learning algorithm on image denoising, and compares it to OMP and the proximal method for sparse coding, as well as the more recently proposed SOUPDIL method [14]; as considered in the latter, the dictionary update method is fixed to the SVD, which allows to provide fair comparable results between all sparse coding methods.

A. Experiments on synthetic data

To illustrate the relevance of the proposed MIQP, experiments are conducted on synthetic data, thus with ground-truth data to assess the sparse coding phase only. We consider the problem posed in (2) of estimating sparse coefficients X from signals Y with a given dictionary D.

In these experiments, a sparse matrix $X \in \mathbb{R}^{128 \times 10000}$ is created with a column-wise maximum sparsity level of 6, by using the K-SVD method applied on a randomly generated matrix $Y_0 \in \mathbb{R}^{64 \times 10000}$. Then, to assure the fairness of the experiments, a new column-wise normalized matrix $D \in \mathbb{R}^{64 \times 128}$ is randomly generated. With the known matrices X and D, the training data Y is finally produced by the following equation

$$Y = DX + \kappa E, \tag{16}$$

where E is a randomly generated zero-mean white Gaussian noise matrix and κ a parameter controlling the noise level, set to $\kappa = 0.01$ in the experiments. For statistical purpose, data of size 10000 is divided into 100 units. For each unit $Y_i \in \mathbb{R}^{64 \times 100}$, a sparse code matrix X_i is estimated. This allows to provide the median, the 5th and 95th percentiles.

The two MIQP sparse coding algorithms, with and without initialization, are compared to OMP and the proximal method presented in Section III-C. The performances of the sparse coding methods are evaluated with three criteria: the difference between X_i and the estimated $\hat{X}_i$, i.e., $\|X_i - \hat{X}_i\|_F$, the reconstruction error, i.e., $\|Y_i - D\hat{X}_i\|_F^2$, and the percentage of zero and non-zero elements being found in the right positions.

Gurobi Optimizer v7.0.2 has been chosen to solve the MIQP optimization problem. We run the software in the Matlab® environment on a server with 4 Intel® Xeon® processors with a CPU clocked at 2.4 GHz. The parameters settings of the Gurobi solver are set using the default values with a time limit of 20 seconds. For the initialization by the proximal method, the number of iterations is set to 200.

TABLE I presents the computational time and results about the reconstruction error, the accuracy of the sparse coding estimation and the percentage of number of the zero and non-zero elements being recovered in the right position. It shows that the errors obtained by MIQP is far less than that the ones of OMP and proximal method. Furthermore, the introduction of initialization has a little effect on the accuracy,

while the computational cost is reduced by a factor of 5. In addition, more non-zero elements are found in the right positions. These advantages make MIQP of great interest to be used as a sparse coding algorithm and, in conjunction with a dictionary updating rule, as a dictionary learning algorithm. In spite of the overall strength of MIQP, its Achilles' Heel is the excessive computational complexity, making it difficult to use for large-scale problems. However, as aforementioned, the proposed acceleration opens the possibility to apply the MIQP-based dictionary learning algorithm on large-scale problems, such as in image denoising. Note that, in practice to get an improvement over the proximal method, there is no need to run the optimization until the global minimum. Whatever the computing budget is allocated, the MIP formulation allows to use it to improve the results.

B. Design of experiments on real images

We choose the segments of natural images in the USC-SIPI Image Database[1] for experiments. The dataset contains five frequently used images in signal and image processing, as presented in Fig. 1. The images are of size 121×121. The images in all experiments are corrupted with an additive zero-mean white Gaussian noise.

The experiments are conducted using two different settings in order to denoise some given corrupted image. The first one, called large-scale dictionary learning, considers a corpus of high-quality images to learn a single dictionary, then uses the obtained dictionary to denoise the corrupted image. The second setting, called adapted dictionary, learns the dictionary from the corrupted image to be denoised. These two settings allow to prove the semantic representation power of the dictionary, that is to say, the atoms in dictionary contain real semantic information. An exact optimization approach is assumed to be resistant to additive noise and can recover the original signals. Consequently, with all the semantic representative atoms in the dictionary, the corresponding signal can be recover directly.

The experiment is conducted with the same Gurobi solver as aforementioned and in the same environment and settings. The parameters settings of Gurobi are: TimeLimit 50 and IterationLimit 500. For initialization by the proximal method, the number of iterations is also set to 200. The coefficient to decrease M is set to $\alpha = 2.5$.

To assess the quality of denoising an image $\widetilde{Y}$, we consider the peak signal-to-noise ratio (PSNR), namely

$$\text{PSNR} = -10 \log \frac{\|\widehat{Y} - \widetilde{Y}\|^2}{255^2},$$

where $\widehat{Y}$ denotes the reconstructed image. The reconstruction model proposed by Elad and Aharon in [7] with

$$\widehat{Y} = \left(\lambda I + \sum_{ij} R_{ij}^T R_{ij}\right)^{-1}\left(\lambda \widehat{Y} + \sum_{ij} R_{ij}^T D\mathbf{x}_{ij}\right), \tag{17}$$

where the matrix R_{ij} is the matrix extracting the (i, j)-th block from the image, and λ is set to $30/\sigma$ as recommended

---

[1]http://sipi.usc.edu/database/database.php?volume=misc

TABLE I: Computational time and accuracy results (100-batch median, 5th and 95th percentiles) on synthetic data

Method / Results	Computational time			Reconstruction error $\|Y_i - D\hat{X}_i\|_F^2$			Sparse coding error $\|\hat{X}_i - X_i\|_F$			Position accuracy of non-zero elements (%)		
	P_5	median	P_{95}	P_5	median	P_{95}	P_5	median	P_{95}	P_5	median	P_{95}
OMP	0.019	0.020	0.032	14.00	15.90	17.45	39.17	44.84	51.13	98.16	98.52	98.81
Proximal	0.023	0.024	0.032	13.20	15.14	16.19	34.68	42.33	46.40	98.42	98.62	98.98
MIQP without initialization	2002.5	2002.6	2002.7	2.36	2.84	**2.86**	5.41	6.45	**6.46**	99.73	99.74	99.78
MIQP with initialization	286.98	415.34	543.90	**1.58**	**2.74**	3.51	**3.60**	**6.06**	7.99	**99.94**	**99.97**	**99.98**

(a) Barbara   (b) Cameraman   (c) Elaine   (d) Lena   (e) Man

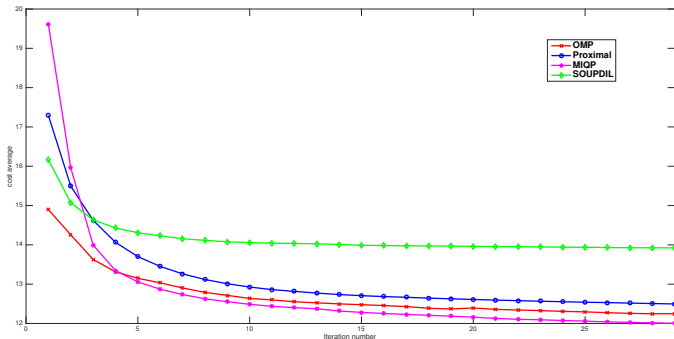Fig. 1: Examples in the USC-SIPI Image Database

Fig. 2: Convergence of the proposed algorithm and the comparison with K-SVD using OMP, proximal method and SOUPDIL

in [7], where the authors demonstrate the superiority of this reconstruction model on the conventional one $\hat{Y} = DX$. This superiority is also observed in the experiments in this paper.

C. Large-scale (global) dictionary learning

The first setting considers the set of high-quality images in order to construct a unique global dictionary that will serve to denoise every image. More than $\ell \approx 1.6 \times 10^4$ overlapping patches of size $n = 8 \times 8$ from the images are extracted to get a single training dataset denoted Y. The number of the atoms is set to $p = 256$ and the sparsity level is $T = 20$

(these parameters are determined by preliminary experiments and corroborated by other studies, such as [7]).

The proposed dictionary learning algorithm (MIQP for sparse coding, SVD for dictionary updating) is executed for 30 iterations to learn a dictionary. This number of iterations is more than enough for convergence, as illustrated in Fig. 2. This figure also shows how the proposed method converges faster than the other dictionary learning methods (K-SVD with OMP, proximal method and SOUPDIL algorithm).

For the purpose of studying the properties of the learned dictionary, the sparsity-constrained formulation (2) and the error-constrained formulation (3) are respectively investigated to learn the sparse code for reconstruction. The sparsity-constrained formulation (2) defines a sparse coding problem with a predefined sparsity parameter T. Considering the error-constrained optimization problem (3), it is easy to make OMP satisfy the constraint by measuring the reconstruction error each time after adding a non-zero entry [7]; The proximal method will search for the Pareto optimal when the sparsity level varies [40]; MIQP keeps all the signals in the constraint based on the decided sparsity of initialization obtained by the proximal method. As recommended in [7], $\epsilon = c\,n\,\sigma^2$ with $c = 1.15$ and a maximum sparsity parameter T_m (usually the same as T) set to assure the sparse level. In order to understand the influence of the noise level on the results of the proposed method, we consider additive Gaussian noise of different standard deviations ($\sigma = 10, 20, 50$ in the experiments).

The reconstruction accuracy is given in TABLE II in terms of the PSNR. These results to evaluate sparse coding show that

TABLE II: Accuracy of the denoising in terms of the PSNR in the large-scale (global) dictionary learning, for each of the five images at several noise levels, comparing the sparsity-constrained formulation (2) and the error-constrained formulation (3) (the higher, the better)

Image	Sparse coding formulation	$\sigma = 10$	$\sigma = 20$	$\sigma = 50$
Barbara	error-constrained	24.71	23.75	20.79
	sparsity-constrained	26.77	25.24	20.14
Cameraman	error-constrained	24.93	23.90	20.16
	sparsity-constrained	27.70	25.75	20.19
Elaine	error-constrained	26.78	25.64	21.57
	sparsity-constrained	29.87	27.81	21.14
Lena	error-constrained	26.05	24.98	21.22
	sparsity-constrained	28.83	26.93	20.92
Man	error-constrained	24.67	23.68	20.80
	sparsity-constrained	27.60	25.97	20.10

Fig. 3: Histogram of the coherence of learned dictionary

the sparsity-constrained formulation (2) always outperforms the error-constrained formulation (3) when $\sigma = 10$ (with an average improvement 2.73dB) and $\sigma = 20$ (with an average improvement 1.95dB). At high noise level with $\sigma = 50$, their performances are comparable.

To measure the quality of the dictionaries, we consider the coherence (correlation measured with the inner product) between the atoms of each dictionary, thus measuring how much two atoms in the dictionary are similar. This fundamental information allows to define more powerful measures, such as the coherence and Babel function [41], [42]. The coherence measure of a given dictionary, defined by the maximum absolute inner product between two distinct atoms, provides strong insights on the capacity of the dictionary to recover sparse signals. For instance, it is shown in [41] that a μ-coherence dictionary can recover a K-sparse signal if $\mu < \frac{1}{2K-1}$. It is well known that the OMP algorithm (e.g. K-SVD) often provides dictionaries with high coherence, and most atoms are highly correlated. To overcome this issue, several strategies have been proposed to provide more incoherent dictionaries (see [43] and references therein). Fig. 3 provides the histogram of the coherence between the atoms of the learned dictionaries, for each of the four methods under investigation. It is observed that the coherence of the obtained dictionary can be ordered as follows

$$\text{SOUPDIL} \prec \text{MIQP} \prec \text{Proximal method} \prec \text{K-SVD}.$$

Besides the analysis of the dictionary quality, we study next the overall performance on the image denoising problem.

D. Adapted dictionary learning

In the second setting, the dictionary is trained on the corrupted image under scrutiny, and then used to denoise it; the dictionary is then "adapted" to the image at hand. As in the first experiment, the signal matrix is created in the same way using overlapping patches. F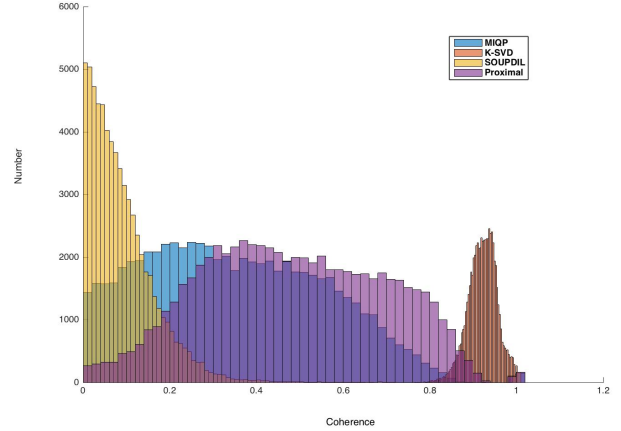or each corrupted image $\widetilde{Y}$, an adapted dictionary is trained on it and then used for denoising the same image.

All three methods, OMP, proximal method and SOUPDIL, are compared with MIQP based dictionary learning method. Moreover, we consider also a variant of K-SVD with OMP, where the signals are pre-centered (subtracting the image mean) prior to learning the dictionary [2]; connections between centered and uncentered data are studied in [44]. In the experiments, SOUPDIL is implemented using the original Matlab code provided by its authors and available here[2]. For the other three methods, the experiments details and the parameter settings are given in the following. When dealing with noisy data in the training phase, the knowledge about the noise level σ is used for restricting the reconstruction error, as shown in the constraint in the optimization problem (3) and the parameter setting $\epsilon = c\,n\,\sigma^2$ with $c = 1.15$. These values, optimized for OMP in [7], are used here for both proximal method and MIQP, thus putting our method in a less favorable situation.

In this part, the error-constrained optimization problem (3) is used for sparse coding. The method of realization is described in the large-scale dictionary learning. In order to ensure the sparsity of the signals, the upper bound T is set to 20 for the proximal and MIQP methods, as in the first setting. By fixing the dictionary updating method to SVD in all the methods, this allows to have a fairly comparable setting to analyze and compare the performance of the sparse coding methods. The number of atoms is set to $p = 256$ for OMP, as suggested in [7] where extensive experiments were conducted. The number of atoms for the proximal method is set to $p = 65$, which is obtained from a set of 14 candidate values $\{50, 55, 60, 65, \ldots, 110, 150, 200, 256, 300\}$ that encloses the most used values in the literature. The same value is used for MIQP, which is a less favorable situation for our method. The total number of iterations is still 30 for the two-step sparse coding and dictionary updating. The SOUPDIL method uses the same parameter setting as recommended in [14] after extensive experimental analysis.

[2]https://gitlab.eecs.umich.edu/fessler/soupdil_dinokat

TABLE III: Denoising results in the adapted dictionary learning setting, for each of the five images, as well as the average results (the higher, the better)

Image	Method	PSNR
Barbara	OMP	22.04
	OMP (pre-centering)	21.97
	proximal	22.54
	SOUPDIL	22.30
	MIQP	**22.59**
Cameramen	OMP	22.54
	OMP (pre-centering)	22.63
	proximal	22.49
	SOUPDIL	**22.79**
	MIQP	22.58
Elaine	OMP	23.00
	OMP (pre-centering)	22.91
	proximal	23.29
	SOUPDIL	**23.43**
	MIQP	23.39
Lena	OMP	22.48
	OMP (pre-centering)	22.51
	proximal	23.08
	SOUPDIL	**23.20**
	MIQP	23.09
Man	OMP	21.23
	OMP (pre-centering)	21.32
	proximal	21.70
	SOUPDIL	21.67
	MIQP	**21.86**

With the the learned dictionaries, the same reconstruction model (17) is used for obtaining the denoised image. TABLE III gives the denoising accuracy in terms of PSNR by using the three aforementioned dictionary learning methods. We notice that the influence of data pre-centering is not always positive. It is observable that MIQP can outperform the K-SVD and proximal methods almost in all cases. On average over all five images, the proposed method carries out an improvement of 0.45 with respect to OMP, and 0.08 with respect to the proximal method. These improvements are important since, on one hand, PSNR is a logarithmic-scale measure and, on the other hand, the parameters were optimized for OMP (*e.g.* ϵ, $c = 1.15$, $p = 256$) and for the proximal method ($p = 65$). Even compared with the state-of-the-art dictionary learning algorithm SOUPDIL, MIQP has comparable performance.

E. On the computational complexity

In despite of the great performance of MIQP method on all images and compared to all the other methods, it has high computational complexity in implementation. Because we have different sizes of the training data in each setting (global dictionary learning and adapted dictionary training for each

image), the training time is not comparable. In the following, we focus on the average time of a single image. While the OMP algorithm and the proximal method require only a couple of minutes for completing the dictionary learning, MIQP needs about one hour. See also TABLE I for results obtained on synthetic data. However, recent advances in MIQP solvers allow to reduce this gap.

Indeed, while the computational complexity remains the Achilles heel of such methods, great improvements are being carried out these days on MIQP solvers. For instance, the new Gurobi Optimizer v8.0, made public a couple of days prior to the submission of this paper, is more than 220% fasteron MIQP problems than the one used in this paper. Moreover, new advances in solvers are exploiting more and more the modern architectures and multi-core processors. Finally, currently available off-the-shelf solvers, such as Gurobi and CPLEX, do not have GPU implementations, which could also provide important computational improvements.

V. CONCLUSION AND FUTURE WORK

In this paper, the K-SVD algorithm was revisited by proposing the exact optimization method MIQP for sparse coding, rather than OMP. Thanks to recent advances in linear programming techniques, as well as more powerful hardware, the speed of computation of MIQP has been greatly improved. Furthermore, by introducing additive constraints and an appropriate initialization, it was proved that it is feasible to use MIQP for sparse coding to redefine the K-SVD algorithm, and apply it in image processing. Though, the MIQP method had much more time complexity in implementation comparing with the approximate methods, the feasibility of the method was proved for large-scale data like well-known images. Moreover, the image denoising experiments showed the advantage of the proposed MIQP method based K-SVD algorithm. Furthermore, the high noise-tolerance of our method was demonstrated on both the large-scale and the adapted dictionary learning settings. Indeed, state-of-the-art methods rely on the approximation of the ℓ_0-norm, while such approximation deteriorates when dealing with noisy data. The resolution of the exact ℓ_0 optimization problem, as proposed in this paper, overcomes this issue.

This paper demonstrated that the exact ℓ_0 optimization problem in dictionary learning can be solved for image processing, working on real images. While having good performance amelioration, its Achilles heel is the computational complexity. However, great improvements are being carried out these days on MIQP solvers, with more than 220% speed enhancement in a single year (*e.g.* Gurobi Optimizer v8.0 versus v7.0).

As for future work, we will address the problem of computational complexity by using recently proposed convex reformulation in [45], as well as other recent developments in linear programming theory. Furthermore, we will extend this work beyond K-SVD to deal with classification, segmentation and object recognition.

REFERENCES

[1] Bruno A. Olshausen and David J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[2] Julien Mairal, Francis Bach, Jean Ponce, et al., "Sparse modeling for image and vision processing," *Foundations and Trends® in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[3] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies, "Image coding using wavelet transform," *IEEE Transactions on image processing*, vol. 1, no. 2, pp. 205–220, 1992.

[4] Laurent Jacques, Laurent Duval, Caroline Chaux, and Gabriel Peyré, "A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity," *Signal Processing*, vol. 91, no. 12, pp. 2699 – 2730, 2011.

[5] Michal Aharon, Michael Elad, and Alfred 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] Karin Schnass, "On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD," *Applied and Computational Harmonic Analysis*, vol. 37, no. 3, pp. 464–491, 2014.

[7] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[8] Avi Ben-Cohen, Eyal Klang, Michal Amitai, and Hayit Greenspan, "Sparsity-based liver metastases detection using learned dictionaries," in *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*. IEEE, 2016, pp. 1195–1198.

[9] Hien Van Nguyen, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa, "Kernel dictionary learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2021–2024.

[10] Yael Yankelevsky and Michael Elad, "Structure-aware classification using supervised dictionary learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4421–4425.

[11] Rodolphe Jenatton, Julien Mairal, Francis R Bach, and Guillaume R Obozinski, "Proximal methods for sparse hierarchical dictionary learning," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 487–494.

[12] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen, "ℓ_0 norm based dictionary learning by proximal methods with global convergence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3858–3865.

[13] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, 2011.

[14] Saiprasad Ravishankar, Raj Rao Nadakuditi, and Jeffrey A Fessler, "Efficient sum of outer products dictionary learning (soup-dil) and its application to inverse problems," *IEEE transactions on computational imaging*, vol. 3, no. 4, pp. 694–709, 2017.

[15] Dimitris Bertsimas, Angela King, Rahul Mazumder, et al., "Best subset selection via a modern optimization lens," *The annals of statistics*, vol. 44, no. 2, pp. 813–852, 2016.

[16] Sébastien Bourguignon, Jordan Ninin, Hervé Carfantan, and Marcel Mongeau, "Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1405–1419, 2016.

[17] M. Rebollo and Laureano F. Escudero, "A mixed integer programming approach to multi-spectral image classification," *Pattern Recognition*, vol. 9, no. 1, pp. 47–57, 1977.

[18] Sergio Garrido-Jurado, Rafael Munoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recogn.*, vol. 51, no. C, pp. 481–491, Mar. 2016.

[19] Xiuzhuang Zhou, Kai Jin, Qian Chen, Min Xu, and Yuanyuan Shang, "Multiple face tracking and recognition with identity-specific localized metric learning," *Pattern Recognition*, vol. 75, pp. 41 – 50, 2018, Distance Metric Learning for Pattern Recognition.

[20] Liujuan Cao, Feng Luo, Li Chen, Yihan Sheng, Haibin Wang, Cheng Wang, and Rongrong Ji, "Weakly supervised vehicle detection in satellite images via multi-instance discriminative learning," *Pattern Recognition*, vol. 64, pp. 417 – 424, 2017.

[21] Christian Bliek, Pierre Bonami, and Andrea Lodi, "Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report," in *Proceedings of the Twenty-Sixth RAMP Symposium*, 2014, pp. 16–17.

[22] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[23] Mehmet Umut Şen and Hakan Erdogan, "Linear classifier combination and selection using group sparse regularization and hinge loss," *Pattern Recognition Letters*, vol. 34, no. 3, pp. 265–274, 2013.

[24] Mila Nikolova, "Relationship between the optimal solutions of least squares regularized with l0-norm and constrained by k-sparsity," *Applied and Computational Harmonic Analysis*, vol. 41, no. 1, pp. 237–265, 2016.

[25] Olivier Bousquet and Léon Bottou, "The tradeoffs of large scale learning," in *Advances in neural information processing systems*, 2008, pp. 161–168.

[26] Deanna Needell and Roman Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 310–316, 2010.

[27] Rahul Garg and Rohit Khandekar, "Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 337–344.

[28] Jérôme Bolte, Shoham Sabach, and Marc Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[29] Julien Mairal and Bin Yu, "Complexity analysis of the LASSO regularization path," in *Proceedings of the 29th International Conference on International Conference on Machine Learning*. Omnipress, 2012, pp. 1835–1842.

[30] Hong Phuong Dang and Pierre Chainais, "Towards dictionaries of optimal size: A bayesian non parametric approach," *Journal of Signal Processing Systems*, pp. 1–12, 2016.

[31] Fei Zhu and Paul Honeine, "Online kernel nonnegative matrix factorization," *Signal Processing*, vol. 131, pp. 143–153, February 2017.

[32] Robert E. Bixby, "A brief history of linear and mixed-integer programming computation," *Documenta Mathematica*, pp. 107–121, 2012.

[33] Robert C. Jeroslow, "There cannot be any algorithm for integer programming with quadratic constraints," *Operations Research*, vol. 21, no. 1, pp. 221–224, 1973.

[34] Arnold Neumaier and Oleg Shcherbina, "Safe bounds in linear and mixed-integer linear programming," *Mathematical Programming*, vol. 99, no. 2, pp. 283–296, 2004.

[35] Oskar von Stryk and Markus Glocker, "Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation," in *Proc. ADPM 2000 - The 4th Int. Conf. on Automatisation of Mixed Processes: Hybrid Dynamical Systems*, 2000, pp. 99–104.

[36] Karla L. Hoffman and Ted K. Ralphs, "Integer and combinatorial optimization," in *Encyclopedia of Operations Research and Management Science*, pp. 771–783. Springer, 2013.

[37] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[38] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[39] Ioan Buciu, "Image denoising via sparse data representation: A comparative study," in *Fundamentals of Electrical Engineering (ISFEE), 2014 International Symposium on*. IEEE, 2014, pp. 1–4.

[40] Charles Soussen, Jérome Idier, Junbo Duan, and David Brie, "Homotopy based algorithms for L0-regularized least-squares," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3301–3316, 2015.

[41] Joel A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[42] Paul Honeine, "Analyzing sparse dictionaries for online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6343–6353, December 2015.

[43] Boris Mailhé, Daniele Barchiesi, and Mark D. Plumbley, "INK-SVD: Learning incoherent dictionaries for sparse representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 3573–3576.

[44] Paul Honeine, "An eigenanalysis of data centering in machine learning," *CoRR*, vol. abs/1407.2904, pp. 1–13, Mar. 2016.

[45] Alain Billionnet, Sourour Elloumi, and Amélie Lambert, "Exact quadratic convex reformulations of mixed-integer quadratically constrained problems," *Mathematical Programming*, vol. 158, no. 1-2, pp. 235–266, 2016.