A Non-convex Relaxation Approach to Sparse Dictionary Learning

Jianping Shi*† Xiang Ren*† Guang Dai† Jingdong Wang‡ Zhihua Zhang†
†Department of Computer Science and Technology, Zhejiang University ‡Microsoft Research Asia
{shijianping5000, renxiangzju, guang.gdai}@gmail.com
jingdw@microsoft.com zhzhang@gmail.com

Abstract

Dictionary learning is a challenging theme in computer vision. The basic goal is to learn a sparse representation from an overcomplete basis set. Most existing approaches employ a convex relaxation scheme to tackle this challenge due to the strong ability of convexity in computation and theoretical analysis. In this paper we propose a non-convex online approach for dictionary learning. To achieve the sparseness, our approach treats a so-called minimax concave (MC) penalty as a nonconvex relaxation of the ℓ_0 penalty. This treatment expects to obtain a more robust and sparse representation than existing convex approaches. In addition, we employ an online algorithm to adaptively learn the dictionary, which makes the non-convex formulation computationally feasible. Experimental results on the sparseness comparison and the applications in image denoising and image inpainting demonstrate that our approach is more effective and flexible.

1. Introduction

Sparse representation [16] is becoming a promising issue in computer vision and pattern recognition. It is reasonable in that most images can be represented with a sparse linear combination of basis elements called atoms. The atoms are usually chosen from an overcomplete dictionary. Particularly, this overcomplete dictionary consists of atoms whose number greatly exceeds the dimension of the image space. Compared with other combinations of atoms, images using sparse representation enable more flexibility to adapt the representation to the data. Thus, it can provide high performance, especially for applications such as image denoising, image compression, image inpainting as well as image classification.

Naturally, the problem of finding a dictionary and its sparse representation with the smallest number of atoms is modeled by using the ℓ_0 penalty. Unfortunately, the resulting problem is usually NP-hard. This inspires us to relax the ℓ_0 penalty into some tractable alternatives.

A widely used approach is to employ the ℓ_1 penalty as a convex surrogate, e.g., the lasso [17]. Many algorithms have been developed to solve this problem, including LARS [18], coordinate-descent algorithms [6], etc. The lasso enjoys some attractive statistical properties. However, the ℓ_1 penalty may bring out over-penalization on some good variables in some cases [15, 20].

A number of non-convex relaxation approaches, such as the log penalty [7], the smoothly clipped absolute deviation (SCAD) penalty [5] and the minimax concave (MC) penalty [20], have been also proposed. In comparison with the log and SCAD penalties, the MC penalty performs the best facing multiple minima and nesting of shrinkage thresholds [15]. Computationally, Breheny *et al.* [4] showed that the coordinate-descent algorithm [6] is very efficient for the MC penalty to find good solutions.

In fact, the MC penalty is tighter than the ℓ_1 penalty [20]. The MC penalty is characterized by a positive factor γ. With $\gamma \to \infty$, it is the ℓ_1 penalty and with $\gamma \to 1+$, it is the ℓ_0 penalty.

In this paper we develop a novel dictionary learning method using the non-convex MC penalty. The main contributions are listed as follows:

1. We treat the MC penalty as a non-convex relaxation of the ℓ_0 penalty for dictionary learning, which results in a more robust and sparse representation.
2. We devise an online algorithm to make the non-convex formulation computationally feasible, and its convergence is theoretically guaranteed.
3. We conduct empirical comparison of our method with the existing dictionary learning method that adopts the ℓ_1 penalty in the applications to image denoising and image inpainting, which further justifies that our approach yields more robust and sparse representation.

The rest of this paper is organized as follows. Section 2 reviews the related work. In Section 3 we discuss the dic-

[1] *Equal contribution

tionary learning problem. In Section 4 we present our online dictionary learning method based on the MC penalty. The applications on image denoising and image inpainting as well as experiment results are reported in Section 5. Finally, we conclude our work in Section 6.

2. Related Work

Dictionary learning problems aim to learn an effective dictionary to adapt specific data, giving rise to a sparse presentation using only a few atoms of the dictionary. The problem is motivated from sparse representation [16] and has recently attracted great interest in the computer vision community.

Some of previous works have focused on sparse coding with a fixed dictionary. Recently, using a linear combination of a few atoms from a *learned* dictionary, instead of a pre-defined one, has received a lot of attention. For example, Aharon *et al.* [2] generalized the K-means clustering process to alternate between sparse coding and dictionary updating. Mairal *et al.* [11] proposed an online model to efficiently solve this problem. Jenatton *et al.* [8] used a tree-structured sparse representation to give a linear-time computation. Mairal *et al.* [12] also developed a discriminative approach, instead of a purely reconstructive approach, to build a dictionary. This supervised manner can be effectively used in applications such as image classification.

In addition, the dictionary learning method has been widely applied in computer vision and pattern recognition. Aharon *et al.* [2] used the dictionary learning model for image denoising by filling in missing pixels and image compression. Yang *et al.* [19] jointly learned two dictionaries for low resolution and high resolution images respectively to deal with an image super-resolution task. Zhang *et al.* [21] generalized the K-SVD algorithm for face recognition by incorporating the classification error.

As we see, all these methods are based on the convex ℓ_1 penalty. In this paper we attempt to use the non-convex MC penalty for addressing the dictionary learning problem.

3. Problem Formulation

Throughout this paper, we present the notation in what follows. For $\mathbf{a} = (a_1, \ldots, a_m)^T \in \mathbb{R}^m$, let $\|\mathbf{a}\|_0$ be the ℓ_0 norm of $\mathbf{a}$ (i.e., the number of the nonzero a_j), $\|\mathbf{a}\|_1 = \sum_{j=1}^m |a_j|$ be the ℓ_1 norm of $\mathbf{a}$, and $\|\mathbf{a}\|_2 = (\sum_{j=1}^m a_j^2)^{1/2}$ be the ℓ_2 norm of $\mathbf{a}$. In addition, let $\mathrm{sgn}(\cdot)$ be the sign function.

Given a signal $\mathbf{x} \in \mathbb{R}^p$, we are now concerned with its sparse approximation over a dictionary $\mathbf{D} \in \mathbb{R}^{p \times q}$, each column of which is referred to an atom. That is, we attempt to find a linear combination of only "few" atoms, which is "close" to the signal $\mathbf{x}$. Note that the dictionary is usually overcomplete, which implies that $q > p$ is possible. To avoid trivial solutions and bring the convergence analysis into existence, $\mathbf{D}$ is restricted to the set $\mathcal{C}$, which is defined as

$$\mathcal{C} \triangleq \{\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_q] : \mathbf{d}_j^T \mathbf{d}_j = 1, \forall j = 1, \ldots, q\}. \quad (1)$$

For a training set of n signals $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, dictionary learning can be formulated as an empirical optimization problem,

$$\min_{\mathbf{D} \in \mathcal{C},\, \mathcal{B}} \frac{1}{n} \sum_{i=1}^n L_i(\boldsymbol{\beta}_i, \mathbf{D}),$$

where $\mathcal{B} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n\}$ and

$$L_i(\boldsymbol{\beta}_i, \mathbf{D}) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + P(\boldsymbol{\beta}_i, \lambda). \quad (2)$$

Here $P(\boldsymbol{\beta}_i, \lambda)$ is a penalty function with $\lambda > 0$ a tuning parameter controlling model complexity.

A sparse representation for $\mathbf{x}_i$ can be obtained by setting some elements of $\boldsymbol{\beta}_i$ zeros. In the dictionary learning problem, a sparsity-inducing norm is usually used as a regularization for the optimization function to obtain sparse solution.

One immediate idea is to define $P(\boldsymbol{\beta}_i, \lambda)$ as the ℓ_0 norm of $\boldsymbol{\beta}_i$; namely, $P(\boldsymbol{\beta}_i, \lambda) = \lambda\|\boldsymbol{\beta}_i\|_0$. However, the resulting optimization problem is usually NP-hard.

Alternatively, as a convex relaxation of the ℓ_0 penalty, the ℓ_1 penalty $P(\boldsymbol{\beta}_i, \lambda) = \lambda\|\boldsymbol{\beta}_i\|_1$ is widely used in the literature [17]. This results in the recent developments of sparse learning in computer vision such as [9, 11].

With the use of the ℓ_1 penalty, the dictionary learning problem is expressed as follows

$$\min_{\mathbf{D} \in \mathcal{C},\, \mathcal{B}} \frac{1}{n} \sum_{i=1}^n \Big\{\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + \lambda\|\boldsymbol{\beta}_i\|_1\Big\}. \quad (3)$$

Noted that it is allowed to take different tuning parameters λ for different penalty functions $P(\boldsymbol{\beta}_i, \lambda)$. For the sake of simplicity, we assume that the same tuning parameters are applied to every penalty function.

To solve the problem in (3), a natural approach is to alternatively optimize between $\mathbf{D}$ and $\mathcal{B}$. That is, minimize one while keeping the other one fixed.

When fixing $\mathbf{D}$, the problem is called *sparse coding* which is the conventional *lasso* model [17]. Thus, it can be efficiently solved by the LARS algorithm [18], the coordinate descent algorithm [6], etc.

When fixing $\mathcal{B}$, we can resort to the classical first-order projected stochastic gradient descent algorithm [1] for estimating $\mathbf{D}$. The algorithm consists of a sequence of updates of $\mathbf{D}$

$$\mathbf{D}_t = \prod_{\mathcal{C}}[\mathbf{D}_{t-1} - \delta_t \nabla_{\mathbf{D}} L_t(\boldsymbol{\beta}_t, \mathbf{D}_{t-1})], \quad (4)$$

where $\mathbf{D}_t$ is the estimation of the optimal dictionary at the tth iteration, $\boldsymbol{\beta}_t$ is the estimation of the sparse code for the signal $\mathbf{x}_t$ at the tth iteration, δ_t is the gradient step, and $\prod_{\mathcal{C}}$ represents the projector to refine the dictionary to the set $\mathcal{C}$.

Recently, based on the stochastic gradient algorithm, Mairal *et al.* [11] proposed an online dictionary learning method, which handles one input signal at a time.

Although the lasso enjoys attractive statistical properties in dictionary learning problem, it might fall short in following situations. First, to induce sparsity, the lasso ends up shrinking the coefficients more for the "good" variables. Furthermore, if these selected "good" variables are strongly correlated, this effect is exacerbated, and may mistakenly include other variables to this model.

In this paper, to overcome these drawbacks, we employ the non-convex MC penalty for the sparse dictionary learning, giving rise to the same or better prediction accuracy and superior variable selection properties in comparison with one based on the lasso.

4. Methodology

The MC penalty family is defined as:

$$\begin{aligned}P(\alpha,\lambda,\gamma) &= \lambda\int_0^{|\alpha|}(1-\frac{z}{\gamma\lambda})_+dz\\ &= \lambda\Big(|\alpha| - \frac{\alpha^2}{2\lambda\gamma}\Big)I(|\alpha|<\lambda\gamma) + \frac{\lambda^2\gamma}{2}I(|\alpha|\geq\lambda\gamma),\end{aligned}$$

where $(u)_+ = \max\{u,0\}$, and $I(u)$ is the indicator function for u. For each $\lambda > 0$, there is a continuum of penalties with respect to γ varying from ∞ to $1+$. Specially, when $\gamma \to \infty$, $P(\alpha,\lambda,\gamma) \to \lambda|\alpha|$, it is exactly a soft-threshold operator as the ℓ_1 norm; when $\gamma \to 1+$, it gives rise to a hard threshold operator corresponding to the ℓ_0 norm. Thus, we assume that $\gamma \in (1,\infty)$ in this paper.

Applying the MC penalty to the dictionary learning problem yields

$$\min_{\mathbf{D}\in\mathcal{C},\mathcal{B}} \Big\{L(\mathcal{B},\mathbf{D};\lambda,\gamma) \triangleq \frac{1}{n}\sum_{i=1}^{n} L_i(\boldsymbol{\beta}_i,\mathbf{D};\lambda,\gamma)\Big\}, \quad (5)$$

where

$$L_i(\boldsymbol{\beta}_i,\mathbf{D};\lambda,\gamma) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + P(\boldsymbol{\beta}_i,\lambda,\gamma). \quad (6)$$

Denote $\boldsymbol{\beta}_i = (\beta_{i1},\ldots,\beta_{iq})^T$ for $i = 1,\ldots,n$. Further, the penalty $P(\boldsymbol{\beta}_i,\lambda,\gamma)$ can be expressed as:

$$P(\boldsymbol{\beta}_i,\lambda,\gamma) = \lambda\sum_{j=1}^{q}\int_0^{|\beta_{ij}|}\Big(1-\frac{z}{\gamma\lambda}\Big)_+dz. \quad (7)$$

4.1. Online Approach

To make our dictionary learning problem scalable, we resort to the online approach given in [11] for solving the optimization problem in (5).

The basic idea of the online learning is to use a surrogate function Q_t to calculate the dictionary in an online manner. In particular, the surrogate function Q_t is:

$$Q_t(\mathcal{B}^{(t)},\mathbf{D}) \triangleq \frac{1}{t}\sum_{i=1}^{t}\Big\{\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + P(\boldsymbol{\beta}_i,\lambda,\gamma)\Big\}, \quad (8)$$

where $\mathcal{B}^{(t)} = \{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_t\}$. Assume that $\mathbf{D}_{t-1}$ is the minimizer of $Q_{t-1}(\mathcal{B}^{(t-1)},\mathbf{D})$. In the online manner, the sparse coding $\boldsymbol{\beta}_t$ corresponding to the new coming data $\mathbf{x}_t$ can be calculated by minimizing $L_t(\boldsymbol{\beta}_t,\mathbf{D};\lambda,\gamma)$ with $\mathbf{D} = \mathbf{D}_{t-1}$.

Then, we employ the routine consisting of two steps to find the solutions of $\mathbf{D}$ and $\mathcal{B}$. The first step updates the $\mathcal{B}$ with the fixed $\mathbf{D}$, leading to sparse codings. The second step updates $\mathbf{D}$ with the fixed $\mathcal{B}$ in an online manner, leading to a dictionary.

4.2. Sparse Coding

When the dictionary $\mathbf{D}$ ($\triangleq [\mathbf{d}_1,\ldots,\mathbf{d}_q] \triangleq [d_{ij}]$) is fixed, we tend to optimize $\mathcal{B}$ to obtain the corresponding sparse codings. Further, the optimization problem (5) can be reduced to n optimization problems, each for the update of one $\boldsymbol{\beta}_i$; that is,

$$\min_{\boldsymbol{\beta}_i}\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + \lambda\sum_{j=1}^{q}\int_0^{|\beta_{ij}|}\Big(1-\frac{z}{\gamma\lambda}\Big)_+dz. \quad (9)$$

According to the literature [4, 15], we can differentiate (9) with respect to β_{ij} and discuss the situation for different values of the parameters. The solution of (9) for $\gamma > 1$ can be obtained by the coordinate-wise method. In particular, given the current update $\tilde{\boldsymbol{\beta}}_i$, the next coordinate-wise updates for the univariate β_{ij} are given by

$$\beta_{ij} = S_\gamma(\sum_{l=1}^{p}(x_{il} - \tilde{x}_{il}^j)d_{lj},\lambda).$$

where $\tilde{x}_{il}^j = \sum_{k\neq j} d_{lk}\tilde{\beta}_{ik}$ and $S_\gamma(\cdot,\cdot)$ is the generalized thresholding operator as follows

$$S_\gamma(z,\lambda) = \begin{cases} 0, & \text{if } |z| \leq \lambda,\\ \mathrm{sgn}(z)(\frac{|z|-\lambda}{1-1/\gamma}), & \text{if } \lambda < |z| \leq \lambda\gamma,\\ z, & \text{if } |z| \geq \lambda\gamma. \end{cases} \quad (10)$$

By observing the above thresholding operator, it is not difficult to find that: when $\gamma \to 1+$, the MC penalty approximates to the ℓ_0 penalty; when $\gamma \to \infty$, the MC penalty approximates to the ℓ_1 penalty. The corresponding algorithm for sparse coding is summarized as Algorithm 1, where the

Algorithm 1 Sparse Coding using MC Penalty

```
Input: input signal $\mathbf{x}_i$, the dictionary $\mathbf{D}$, a regularization parameter $\lambda_0$, an increasing array $\gamma = [\gamma_0, \gamma_1, \ldots, \gamma_M]$.
Initialize: $\boldsymbol{\beta}_i(\gamma_{M+1}, \lambda_0) = (\beta_{i1}, \ldots, \beta_{iq})' = \mathbf{0}$.
for $h = M$ to 0 do
    $\tilde{\boldsymbol{\beta}}_i \leftarrow \boldsymbol{\beta}_i(\gamma_{h+1}, \lambda_0)$.
    repeat
        Cycle through $j$ to update $\tilde{\beta}_{ij}$:
            $\tilde{\beta}_{ij} = S_{\gamma_h}(\sum_{l=1}^{p}(x_{il} - \tilde{x}_{il}^{j})d_{lj}, \lambda_0),$
        where $\tilde{x}_{il}^{j} = \sum_{k \neq j} d_{lk}\tilde{\beta}_{ik}$.
        $\boldsymbol{\beta}_i(\gamma_h, \lambda_0) \leftarrow \tilde{\boldsymbol{\beta}}_i$.
    until convergence
end for
Return $\boldsymbol{\beta}_i(\gamma_0, \lambda_0)$
```

sparse code approximating to the ℓ_0 penalty can be calculated using an increasing array $\gamma = [\gamma_0, \ldots, \gamma_M]$.

It is worth pointing out that although the MC penalty $P(\boldsymbol{\beta}_i, \lambda, \gamma)$ is non-convex for $\boldsymbol{\beta}_i$, $L_i(\boldsymbol{\beta}_i, \lambda, \gamma)$ might be convex for β_{ij} under certain mild conditions [4]. Thus, we can also establish the convergence property of our coordinate descent algorithm for the sparse coding based on the MC penalty, which is similar to that in [4].

Lemma 1 *Let $L_i(\boldsymbol{\beta}_i, \mathbf{D}; \lambda, \gamma)$ in (6) be the function of a single variable β_{ij} and keep other variables fixed. If $\gamma > 1$ and $\mathbf{d}_j^T\mathbf{d}_j = 1$ holds, then L_i is a convex function of β_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, q$.*

Using Lemma 1, we can establish the convergence property of our coordinate descent algorithm for the sparse coding based on the MC penalty, and the detailed convergence of the coordinate-wise update can be found in Theorem 1, which can be immediately obtained from [4].

Theorem 1 *Assume that the dictionary $\mathbf{D}$ is given. Let $\{\boldsymbol{\beta}_i^1, \boldsymbol{\beta}_i^2, \boldsymbol{\beta}_i^3, \ldots\}$ denote the sequence of sparse codings produced at each iteration of our coordinate descent algorithms for the signal $\mathbf{x}_i (i = 1, \ldots, n)$. For all $t = 1, 2, 3, \ldots$*

$$L_i(\boldsymbol{\beta}_i^{(t+1)}, \mathbf{D}; \lambda, \gamma) \leq L_i(\boldsymbol{\beta}_i^{(t)}, \mathbf{D}; \lambda, \gamma).$$

In addition, the sequence is guaranteed to converge to a point that is both a local minimum and a global coordinate-wise minimum of $L_i(\boldsymbol{\beta}_i, \mathbf{D}; \lambda, \gamma)$.

4.3. Dictionary Learning

When $\boldsymbol{\beta}_i$ are fixed, we tend to optimize the dictionary $\mathbf{D}$ using the online Newton's method [11].

Algorithm 2 Dictionary Learning Algorithm

```
Input: input signals $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, a regularization parameter $\lambda_0$, an increasing array of $\gamma = [\gamma_0, \gamma_1, \ldots, \gamma_M]$, a initial dictionary $\mathbf{D}_0$.
Initialize: $\mathbf{A}_0 = \mathbf{0}(\in \mathbf{R}^{p \times p})$ and $\mathbf{B}_0 = \mathbf{0}(\in \mathbf{R}^{p \times q})$.
for each $\mathbf{x}_t$ in $\mathcal{X}$ do
    Sparse coding for the MC penalty using $\gamma$:
        $\hat{\boldsymbol{\beta}}_t = \underset{\boldsymbol{\beta}_t}{\text{argmin}} \; L_t(\boldsymbol{\beta}_t, \mathbf{D}_{t-1}; \lambda_0, \gamma_0).$
    $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \hat{\boldsymbol{\beta}}_t\hat{\boldsymbol{\beta}}_t^T$.
    $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \mathbf{x}_t\hat{\boldsymbol{\beta}}_t^T$.
    Using $\mathbf{D}_{t-1} = [\mathbf{d}_1, \ldots, \mathbf{d}_q]$ as a warm start for $\mathbf{D}_t$.
    repeat
        for j = 1 to q do
            Update $\mathbf{d}_j$ by minimizing (11):
                $\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_t(j,j)}(\mathbf{B}_t(:,j) - \mathbf{D}_{t-1}\mathbf{A}_t(:,j)) + \mathbf{d}_j,$
            if $\|\mathbf{u}_j\|_2 \neq 0$ then
                    $\mathbf{d}_j \leftarrow \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|_2}.$
            end if
        end for
    until convergence
end for
Return $\mathbf{D}_n$.
```

Concretely, we can formulate $\mathbf{D}_t$ in the online version as:

$$\begin{aligned} \mathbf{D}_t &= \underset{\mathbf{D} \in \mathcal{C}}{\text{argmin}} \; Q_t(\mathcal{B}^{(t)}, \mathbf{D}) \\ &= \underset{\mathbf{D} \in \mathcal{C}}{\text{argmin}} \; \frac{1}{t}(\frac{1}{2}\text{tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}_t) - \text{tr}(\mathbf{D}^T\mathbf{B}_t)), \end{aligned} \tag{11}$$

where $\mathbf{A}_t = \sum_{l=1}^{t} \boldsymbol{\beta}_l\boldsymbol{\beta}_l^T$ and $\mathbf{B}_t = \sum_{l=1}^{t} \mathbf{x}_l\boldsymbol{\beta}_l^T$. According to (11), we only need to calculate the corresponding matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ at the tth iteration. Furthermore, we can solve the optimization problem (11) with respect to each jth column $\mathbf{d}_j$ of $\mathbf{D}$, while keeping others fixed. Accordingly, we can find the solution $\mathbf{D}_t$ to achieve the update of $\mathbf{D}$.

The corresponding procedure is summarized in Algorithm 2. This algorithm has two advantages: 1) Unlike the classical first-order projected gradient descent algorithm, it does not need to select the tuning rate; 2) It does not need to store all $\mathbf{x}_i$ and $\boldsymbol{\beta}_i$, leading to less memory requirement. In fact, we only need to update the matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ at each iteration.

Figure 1. Some sample images for our experiments

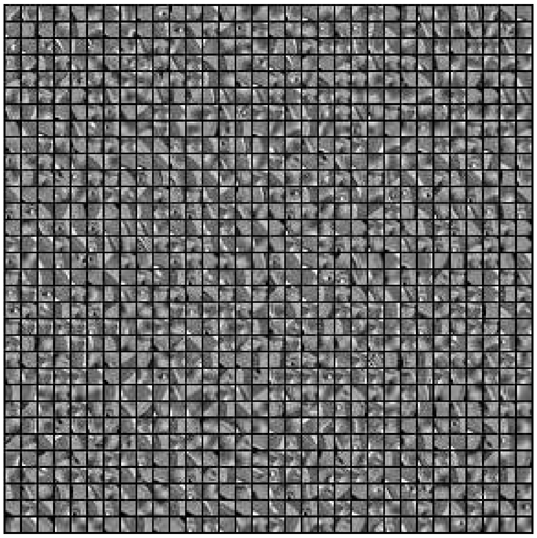


Figure 2. An illustration of the dictionary trained by the generic image data set

5. Experiments

In this section, we conducted several experiments of image denoising and inpainting to evaluate the performance of our proposed dictionary learning method. The experiments were implemented on images from the Berkeley segmentation dataset [14], and some sample images are shown in Fig. 1. The patches in each image, of size 8×8, are regularly sampled in an overlapping manner. Without any loss of generalization, each patch is then normalized to have zero mean. We initialize the MC penalty on 15 different values of γ. They were equally spaced in the range [1.01, 5×10^4] under the log scale, giving an effective approximation for the ℓ_0 penalty. λ is empirically fixed as 0.3 to achieve a better sparsity. The learned dictionary is illustrated in Fig. 2. For the sake of comparison, we also implemented the closely related dictionary learning method based on the ℓ_1 penalty, and the matlab code can be obtained by utilizing the SLEP package [10].

5.1. Illustration

Firstly, we conducted empirically convergence analysis of Algorithm 1. For the more detailed purpose, a set of γ, $\{\gamma_1, \ldots, \gamma_{50}\}$, are equally chosen from $[1.01, \ldots, 5 \times 10^4]$ under the log scale. Fig.3 demonstrates that Algorithm 1 converges with respect to the variation of γ.

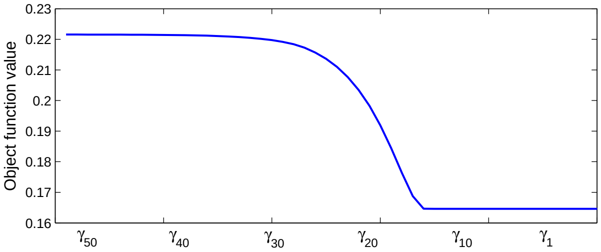


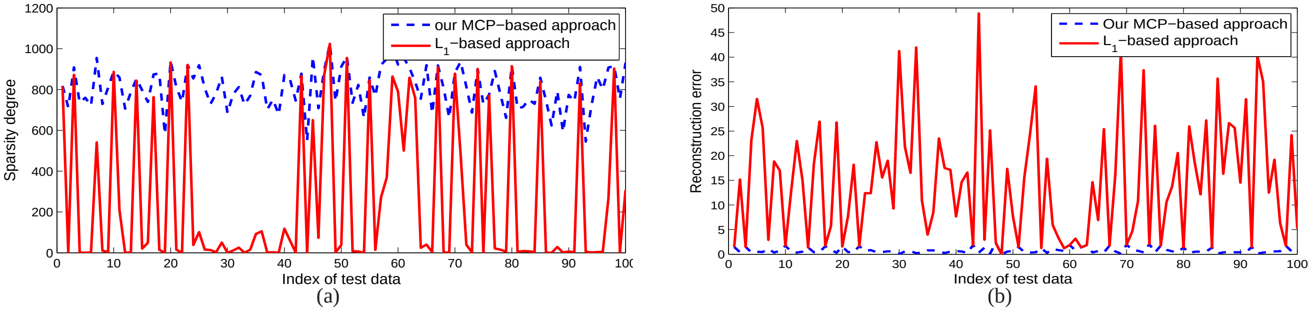
Figure 3. An illustration of the convergence of Algorithm 1 with respect to the variation of γ.

In order to reveal the effectiveness of the MC penalty, we now compare our MC penalty-based dictionary learning approach with the ℓ_1 penalty-based dictionary learning approach in terms of the sparsity degree and the reconstruction (regression) error. It should be pointed out that the sparsity degree is computed as the number of zero components in the sparse reconstruction coefficients $\boldsymbol{\beta}_i$ for images patch $\mathbf{x}_i$. The reconstruction error, corresponding to the vertical axis, is calculated by a typical measurement for regression models, i.e., the root-mean-square error. The horizontal axis corresponds to the patch indices of 100 patches randomly sampled from test images. In the interests of fairness, dictionaries with 1024 atoms used by two approaches were trained corresponding to MC and ℓ_1 penalties. Then, based on the learned dictionary, we did sparse coding with corresponding penalties and evaluate the corresponding sparsity degrees and the reconstruction errors on the test dataset.

The comparative results are depicted in Fig. 4, where Fig. 4(a) shows the sparsity degrees under the almost same reconstruction errors for both, and Fig. 4(b) shows the reconstruction errors under the almost same sparsity degrees for both. It is obvious that: compared with the ℓ_1 penalty approach, our approach can give rise to a higher sparsity degree under the almost same reconstruction errors, or a less construction error under the almost same sparsity degrees; As well, the performance of our approach tends to be more stable with respect to different test images. As a result, the MC penalty can achieve a better performance for the dictionary learning approach in real-world applications.

Table 1. Comparison of the PSNR results on the image denoising. For each image, the left and right columns are the results of our approach and the ℓ_1 penalty approach, respectively.

σ	castle		bridge		horse		farmer		kangaroo	
30	**26.5333**	25.7155	**26.0451**	25.1628	**23.6852**	23.6732	**26.3230**	25.8353	**25.9262**	25.3812
40	**24.3321**	24.0200	**22.1159**	21.4820	**25.7272**	25.5130	**24.2604**	23.9207	**25.1336**	24.8607
50	**23.8084**	23.4974	**20.4872**	20.1762	**24.8706**	23.8727	**24.0685**	23.2993	**24.4902**	23.5519
60	**22.6235**	22.3747	**20.9136**	20.8735	**24.2528**	23.7055	**23.1118**	22.6512	**23.9902**	23.4120



(a)

(b)

Figure 4. Comparison of two dictionary learning methods based on the MC and ℓ_1 penalties: (a) sparsity degree; (b) reconstruction error.

5.2. Color Image Denoising

In the following experiments, we studied the performance of our approach on the color image denoising. Following the setting in [13], we generated noisy images using white Gaussian noise associated with a spatially uniform deviation σ. The scheme of removing the noise is presented as follows. We first regularly sampled the overlapped patches from the noisy image, and computed the sparse reconstruction coefficients with the learnt dictionary. The coefficients were obtained by minimizing $\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i\|_2^2 + P(\boldsymbol{\beta}_i, \lambda, \gamma)$. It should be mentioned here that by considering three different color spaces for each image, the dictionary was jointly learned. In the learning process we regarded a color patch of size 8×8 as a combined vector of $3 \times 8 \times 8$. Then, we reconstruct each patch with learned sparse coefficients, and further reconstruct the whole image by packing the reconstructed patches together to form a new denoised image. For overlapped pixels, we averaged the color value over them.

Fig. 6 shows the visual results for three selected images. We can observe that our dictionary learning approach can provide a better denoising result compared with the ℓ_1 penalty approach on the whole. Specially, our results tend to be much smoother with less artifact. Furthermore, for the castle image, our result with $\sigma = 30$ is comparable to the result with $\sigma = 25$ in [13]. In order to further give a more detailed evaluation for denoising, we also calculate the peak signal-to-noise ratio (PSNR). Table 1 summaries the corresponding results. From Table 1, we can easily see that our approach achieves a better performance of denoising in terms of PSNR. As well, we obtained this improvement under all values of σ and all types of images in the experiment, owing to the robustness of the MC penalty.

5.3. Image Inpainting

In this subsection, we present another image restoration application, i.e., image inpainting. This task basically aims to fill in the missing areas, e.g., removing the occlusions, such as text, subtitles, stamps, and publicity from images. Similar to [13], our approach is not necessarily an efficient model for filling large holes, since a large amount of time and memory would be taken to compute a highly redundant dictionary. Therefore, we compare the performance of filling the holes that are mainly occluded by texts or that are generated by randomly removing some pixels.

For the sake of fairness and simplicity, we implement all compared dictionary learning approaches with 1024 atoms on the Berkeley segment dataset. In the image reconstruction process, we only consider patches that have missing pixels. Reconstruction coefficients of the damaged patches are calculated by minimizing $\frac{1}{2}\|\mathbf{m} \odot (\mathbf{x}_i - \mathbf{D}\boldsymbol{\beta}_i)\|_2^2 + P(\boldsymbol{\beta}_i, \lambda, \gamma)$, where $\mathbf{m}$ is a mask vector to indicate whether the pixel should be removed. The symbol $\odot$ means the element-wise multiplication of two vectors.

The first example is about a widely-used image inpainting example of text removal, which is presented in Fig. 5. It can be observed that our approach, whose result is shown in Fig. 5(c), outperforms the ℓ_1 penalty approach as well as [3]. Fig. 5(d) presents the result based on the ℓ_1 penalty and Fig. 5(e) shows the result in [3]. It should be also noted that the inpainting algorithm in [3] is considered as one of most popular and effective inpainting approaches. To reveal the performance clearly, we zoom in the difficult region, and show them in the lower-right corner of the image. It can

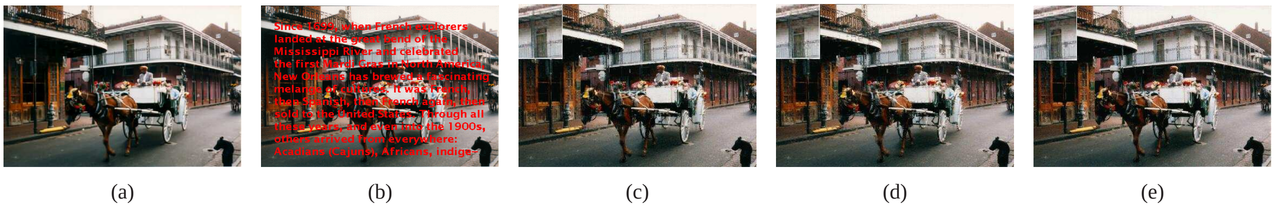

Since 1699, when French explorers
landed at the great bend of the
Mississippi River and celebrated
the first Mardi Gras in North America,
New Orleans has brewed a fascinating
melange of cultures. It was French,
then Spanish, then French again, then
sold to the United States. Through all
these years, and even into the 1900s,
others arrived from everywhere:
Acadians (Cajuns), Africans, indige-
(a)
(b)
(c)
(d)
(e)

Figure 5. Comparison of text removal. (a) original image; (b) image with overlapped texts; (c), (d) and (e) correspond to the inpainting results from our approach, the ℓ_1 penalty approach and [3]

be observed that unlike two other approaches, our approach dose remove the useless error-produced spot and restore the details in the eave. Furthermore, our approach possesses a more excellent ability in presenting the features of objects, such as the outline of the top-middle eave in Fig. 5.

Next, we show a more challenging example in Fig. 7 to evaluate the performance of our approach. In this experiment, we randomly remove 50% pixels of the castle image to form a damaged image, and our task is to fill in the removed pixels. Obviously, the results in Fig. 7 tell us that compared with the ℓ_1 penalty approach [13], our approach performs quite well. In particular, the inpainting performance on the detailed region of the castle image is competitive with that in [13]. Moreover, our result on some flat areas such as the blue sky and the lake surface is much better. Finally, we show the inpainting results of other different images in Fig.8. These images are damaged by various text layers, which have different thickness degrees of words and different expansion areas of layers. Results in Fig. 8(c) are restored image produced by our approach, while those in Fig. 8(d) are produced by ℓ_1 penalty approach. It can be seen that the overall performance of our approach is better than that of the ℓ_1 penalty approach.

6. Conclusion

In this paper we have proposed a non-convex relaxation approach for dictionary learning problems. In particular, we have exploited the MC penalty as a non-convex relaxation of the ℓ_0 penalty in building our model. We have devised a coordinate descend algorithm for solving the model in an online framework, making our non-convex formulation computationally feasible. We have illustrated the applications of our model on image denoising and inpainting. Experimental results have revealed that our approach outperforms the previous method based on the convex ℓ_1 penalty.

Our work shows that non-convex relaxation is a potential approach to sparse learning, with applications in computer vision. Our dictionary learning model is under an unsupervised setting. It would be also interesting to apply non-convex relaxation approaches to semi-supervised or supervised dictionary learning. We will investigate these issues in our future work.

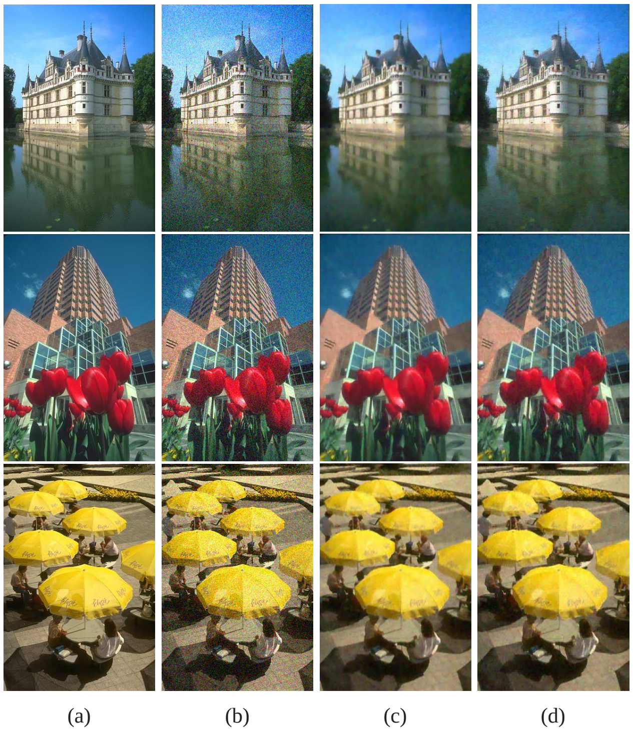


Figure 6. Denoising results when $\sigma = 30$. (a) original images; (b) noisy images; (c) and (d) correspond to the restored images by our approach and the ℓ_1 penalty approach.

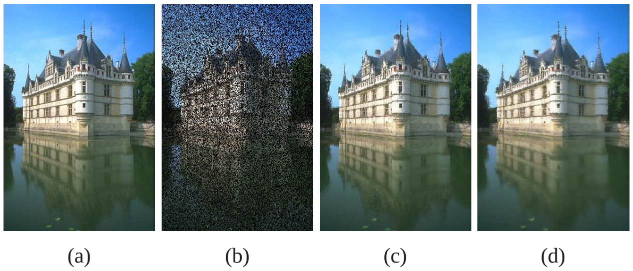
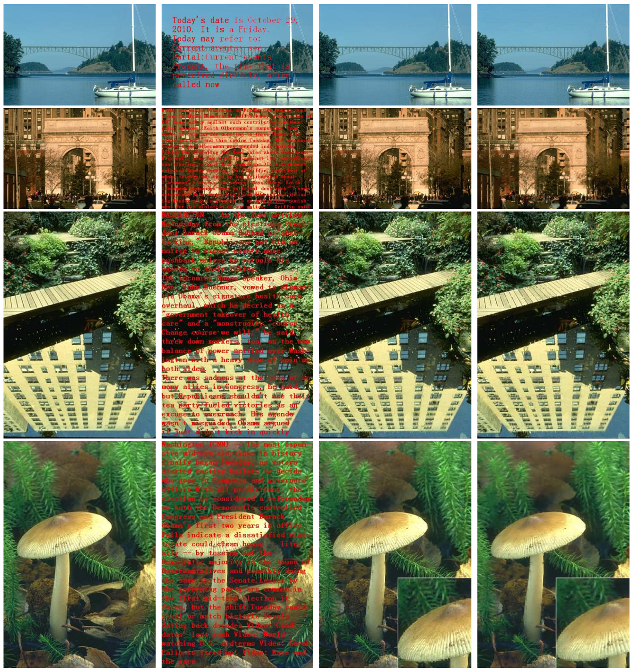


Figure 7. Image inpainting for the image with 50% pixels removed. (a) original image; (b) damaged image; (c) and (d) correspond to the results from our approach and the ℓ_1 penalty approach.



(a) (b) (c) (d)

Figure 8. Image inpainting for text removal. (a) original images; (b) test images; (c) and (d) correspond to the results from our approach and the ℓ_1 penalty approach.

Acknowledges

Zhihua Zhang acknowledges support from the Natural Science Foundations of China (No. 61070239), the 973 Program of China (No. 2010CB327903), the Doctoral Program of Specialized Research Fund of Chinese Universities (No. 20090101120066), and the Fundamental Research Funds for the Central Universities.

References

[1] M. Aharon and M. Elad. Sparse and redundant modeling of image content using an image-signature dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, July 2008. 1810

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006. 1810

[3] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, pages 417–424, 2000. 1814, 1815

[4] P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *To appear in Annals of Applied Statistics*. 1809, 1811, 1812

[5] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001. 1809

[6] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007. 1809, 1810

[7] J. H. Friedman. Fast sparse regression and classification. Technical report, Stanford University, 2008. 1809

[8] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on Machine Learning*, 2010. 1810

[9] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*. MIT Press, 2007. 1810

[10] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009. 1813

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010. 1810, 1811, 1812

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*. MIT Press, 2009. 1810

[13] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008. 1814, 1815

[14] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley, Jan 2001. 1813

[15] R. Mazumder, J. Friedman, and T. Hastie. Sparsenet: Coordinate descent with non-convex penalties. Technical report, Department of Statistics, Stanford University, 2009. 1809, 1811

[16] B. A. Olshausen and D. J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997. 1809, 1810

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996. 1809, 1810

[18] B. E. Trevor, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2002. 1809, 1810

[19] J. Yang, J. Wright, T. Huang, and Y. Ma. Image superresolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010. 1810

[20] C. H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38:894–942, February 2010. 1809

[21] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1810