Simultaneous Sparse Dictionary Learning and Pruning

Simeng Qu and Xiao Wang

Department of Statistics, Purdue University

May 26, 2016

Abstract

Dictionary learning is a cutting-edge area in imaging processing, that has recently led to state-of-the-art results in many signal processing tasks. The idea is to conduct a linear decomposition of a signal using a few atoms of a learned and usually over-completed dictionary instead of a pre-defined basis. Determining a proper size of the to-be-learned dictionary is crucial for both precision and efficiency of the process, while most of the existing dictionary learning algorithms choose the size quite arbitrarily. In this paper, a novel regularization method called the Grouped Smoothly Clipped Absolute Deviation (GSCAD) is employed for learning the dictionary. The proposed method can simultaneously learn a sparse dictionary and select the appropriate dictionary size. Efficient algorithm is designed based on the alternative direction method of multipliers (ADMM) which decomposes the joint non-convex problem with the non-convex penalty into two convex optimization problems. Several examples are presented for image denoising and the experimental results are compared with other state-of-the-art approaches.

1 Introduction

Sparse coding which represents a signal as a sparse linear combination of a representation basis in a dictionary has been successfully applied in many signal processing tasks, such as image restoration [6, 21], image classification [20, 22], to name a few. The dictionary is crucial and plays an important role in the success of sparse representation. Most of the compressive sensing literature takes off-the-shelf bases such as wavelets as the dictionary [4, 5]. In contrast, dictionary learning assumes that a signal can be sparsely represented in a learned and usually over-completed dictionary. The pre-specified dictionary might be universal but will not be effective enough for specific task such as face recognition[24, 11]. Instead, using the learned dictionary has recently led to state-of-the-art results in many practical applications, such as denoising [6, 17, 27, 1], inpainting [13, 15, 18], and image compression [3].

In this paper, we propose a novel regularization method called the Grouped Smoothly Clipped Absolute Deviation (GSCAD) to learn a sparse dictionary and select the appropriate dictionary size simultaneously. It should be emphasized that determining a proper size of the to-be-learned dictionary is crucial for both precision and efficiency of the process. There are not too many existing work on discussing the selection of the dictionary size while most algorithms fix the number of atoms in the dictionary. In general, a two-stage procedure may be used to infer the dictionary size by first learning a dictionary with a fixed size then defining a new objective function penalizing the model complexity [8]. The Bayesian technique can be also employed by putting a prior on the dictionary size [26]. In

addition, many methods have addressed the group variable selection problem in statistics literature [23, 25, 10, 28, 9].

This paper makes four main contributions:

- Our approach imposes sparsity-enforcing constraints on the learned atoms, which improves interpretability of the results and achieves variable selection in the input space.
- Our approach is a one-stage procedure to learn a sparse dictionary and the dictionary size jointly.
- Our proposed algorithm is based on the alternative direction method of multipliers (ADMM) [2]. The joint non-convex problem with the non-convex penalty is decomposed into two convex optimization problems.
- Compared with other state-of-the-art dictionary learning methods, GSCAD has better or competitive performance in various denoising tasks.

2 GSCAD penalty

Review of the Smoothly Clipped Absolute Deviation (SCAD) penalty. SCAD penalty is first proposed by [7] in the context of high dimensional linear regression. SCAD has some desired properties: (i) Unbiasedness: the resulting estimator is nearly unbiased when the true unknown parameter is large; (ii) Sparsity: The resulting estimator is able to sets small estimated coefficients to zero to reduce model complexity; (iii) Continuity: The resulting estimator is continuous in data to avoid instability in model prediction. Defined as

$$\psi_\lambda(d) = \begin{cases} \lambda|d|, & if\ |d| \le \lambda \\ -\frac{|d|^2 - 2c\lambda|d| + \lambda^2}{2(c-1)}, & if\ \lambda < |d| \le c\lambda \\ \frac{(c+1)\lambda^2}{2}, & if\ |d| > c\lambda \end{cases}, \tag{1}$$

for some $\lambda > 0$ and $c > 2$, the SCAD contains three segments. When d is small (less than λ), it acts exactly like the Lasso penalty; when d is big (greater than 3λ), it becomes a constant so that no extra penalty is applied to truly significant parameters; these two segments are connected by a quadratic function which results in a continuous differentiable SCAD penalty function $\psi_\lambda(\cdot)$.

GSCAD penalty. Even though the SCAD penalty possesses many good properties, it only treats parameters individually and does not address any group effect among parameters. With respect to the structure of the dictionary, we propose a new penalty, GSCAD, where G stands for group. Let θ be a vector in $\mathbb{R}^m$. The GSCAD penalty is defined as

$$\Psi_\lambda(\theta) = \log\{1 + \sum_{k=1}^{m} \psi_\lambda(\theta_k)\},$$

where ψ_λ is the SCAD penalty defined in (1). It inherits all three merits of SCAD, unbiasedness, sparsity and continuity, and at the same time takes into account both individual parameters and group effect among parameters. Individually, the GSCAD penalty tends to set small estimated θ_k to zero. Group-wise, if all elements in θ are small, the penalty will penalize the entire vector θ to zero. In addition, if some of the θ_k is significantly large, the penalty will have more tolerance of smaller elements appearing in θ.

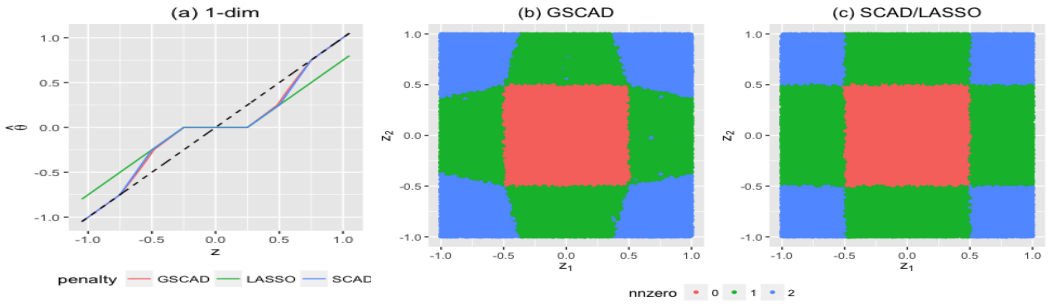

Figure 1: (a) 1-dim threshold function. (b)-(c) Partitions of the 2-dim space $(z_1, z_2) \in \mathbb{R}^2$ according to the number of nonzero elements in $\hat{\theta}$.

To better understand GSCAD, let us consider a penalized least squares problem with an orthogonal design

$$\frac{1}{2}\|z-\theta\|_2^2 + p_\lambda(|\theta|),$$

where z and θ are vectors in $\mathbb{R}^m$. For GSCAD, SCAD and LASSO, the penalty $p_\lambda(|\theta|)$ is, respectively,

$$p_\lambda(|\theta|) = \log\{1+\sum_{k=1}^m \psi_\lambda(\theta_k)\}, \quad p_\lambda(|\theta|) = \sum_{k=1}^m \psi_\lambda(\theta_k), \quad p_\lambda(|\theta|) = \sum_{k=1}^m |\theta_k|.$$

Estimators of θ when $m=1$ are shown in Figure 1 (a), where GSCAD performs very similar to SCAD. All three penalties shows sparsity properties since they all set $\hat{\theta}$ to zero when $|z| \le \lambda$. While the soft-thresholding from LASSO has the inherent bias issue, SCAD and GSCAD give $\hat{\theta} = z$ when $|z| \ge c\lambda$ and and avoid bias. In a two-dimensional case when $m=2$ and $z=(z_1, z_2)$, we investigate partitions of the space according to the number of non-zero element in the resulting estimator $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2)$, see Figure 1 (b)-(c). While SCAD and Lasso treat each coordinate individually, GSCAD takes into account the whole group. It is less likely to set the estimator of one coordinate to zero as the estimator of another coordinate gets away from zero.

Convexity. Even though GSCAD is build upon the non-convex penalty function SCAD, our development uncovers a surprising fact that the optimization problem of GSCAD under orthogonal design is a convex problem. This will greatly facilitates the implementation of GSCAD.

Theorem 1. *Define* $\hat{\theta} = (\hat{\theta}_1, ..., \hat{\theta}_m)$ *as the minima of optimization problem*

$$\min_{\theta\in\mathbb{R}^m} \frac{\varrho}{2}\sum_{k=1}^m (z_k-\theta_k)^2 + \log\{1+\sum_{k=1}^m \psi_\lambda(\theta_k)\}, \quad \textit{with constant } \varrho > 0. \tag{2}$$

Then,

(1) $sign(\hat{\theta}_k) = sign(z_k)$, *and* $|\hat{\theta}_k| \le |z_k|$. *Denote* $\tilde{K} = \{1 \le k \le K : \ z_k \ne 0\}$, *and let* Θ_k *be the open interval between* z_k *and 0. Then problem* (2) *is equivalent to*

$$\min_{\theta_k\in\Theta_k\cup\{0\}, k\in\tilde{K}} \frac{\varrho}{2}\sum_{k\in\tilde{K}} (z_k-\theta_k)^2 + \log\{1+\sum_{k\in\tilde{K}} \psi_\lambda(\theta_k)\} \tag{3}$$

(2) Let $c_0 = card(\tilde{K})$, *be the number of non-zero element in* z. *If*

$$\lambda^2 \le \varrho c_0^{-1} \quad \textit{and} \quad (c-1)\{\varrho(1+\lambda^2)^2 - c_0\lambda^2\} \ge 1+\lambda^2, \tag{4}$$

then optimization problem (3) *is convex, and* $\hat{\theta}$ *is continuous in data* z.

Remarks on Theorem 1. (i) Adding a constant ϱ in (2) makes the problem more general such that the convexity result can be directly applied to the algorithms in Section 3.3, where ϱ plays a role of penalty parameter in the Augmented Lagrangian method. (ii) Condition (4) can be satisfied easily under a wide range of circumstances. For instance, in the previous two-dimensional example with $\varrho = 1$, $c_0 = 2$, and $c = 3$, Condition (4) will be satisfied as long as $\lambda \leq 2^{-1/2}$.

3 Dictionary Learning with GSCAD

3.1 Matrix Factorization Framework

Dictionary learning problems are commonly specified under the framework of matrix factorization. Consider a vectorized clean signal $\mathbf{x} \in \mathbb{R}^m$ and a dictionary $\mathbf{D} = (\mathbf{d}_1, ..., \mathbf{d}_p) \in \mathbb{R}^{m \times p}$, with its p columns referred to as atoms. Sparse representation theory assumes that signal $\mathbf{x}$ can be well approximated by a linear combination of a few atoms in $\mathbf{D}$, i.e.

$$\mathbf{x} \approx \mathbf{D}\alpha,$$

where the number of non-zero elements in α is far less than the number of atoms m. In most of the cases, the clean signal $\mathbf{x}$ won't be available, and instead, we will only be able to observe a noisy signal $\mathbf{y} = \mathbf{x} + \epsilon$, where ϵ represents noise with mean zero and variance σ^2. Suppose we have n signals $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_n) \in \mathbb{R}^{m \times n}$, and we want to retrieve the corresponding clean signals $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$. This can be summarized as a matrix factorization model

$$\mathbf{Y} = \mathbf{D}\mathbf{A} + \epsilon,$$

where $\mathbf{A} = (\alpha_1, ..., \alpha_m)$. To make the problem identifiable, we require the dictionary $\mathbf{D}$ belongs to a convex set $\mathcal{D}$

$$\mathcal{D} = \{\mathbf{D} \in \mathbb{R}^{m \times p} \ s.t. \ \forall j = 1, ..., p, \ ||\mathbf{d}_j||_\infty \leq 1\}.$$

Dictionary learning aims to obtain estimations of dictionary $\hat{\mathbf{D}}$ and sparse coding $\hat{\mathbf{A}}$, and then reconstruct the clean signal as $\hat{\mathbf{x}} = \hat{\mathbf{D}}\hat{\mathbf{A}}$. This is usually done by minimizing the total squared error:

$$\min ||\mathbf{Y} - \mathbf{D}\mathbf{A}||_F^2, \quad \text{subject to additional sparsity constrains on } \alpha,$$

where $||\cdot||_F$ is the Frobenius norm. Constrains such as $||\alpha||_0 \leq L$ (l_0-penalty) and $||\alpha||_1 \leq \lambda$ (Lasso penalty) for some positive constants L and λ are widely adopted by dictionary learning literature. Experiments have shown that Lasso penalty provides better results when used for learning the dictionary, while l_0 norm should always be used for the final reconstruction step [12].

3.2 Regularization on Dictionary

Compared with sparse coding, regularization on dictionary size is less studied. Most of the existing methods, such as K-SVD and Online Learning, estimate the dictionary directly with a fixed dictionary size. They usually require the size of the dictionary to be specified before learning, and this will end up with a solution of over completed dictionary with $p > m$, which may not be very helpful if we want to better understand the mechanism. In addition, learning a sparse dictionary can lower the model complexity and improve interpretability of the results. All these issues can be addressed with the help of GSCAD

penalty, where we would be able to reveal the real size of the dictionary and at the same time obtain an estimated sparse dictionary. More specifically, denote dictionary as $\mathbf{D}$ with p atoms $\mathbf{d}_i = (d_{i1}, \ldots, d_{im})^T \in \mathbb{R}^m, 1 \leq i \leq p$. The GSCAD penalty on dictionary $\mathbf{D}$ is defined by

$$\Psi_\lambda(\mathbf{D}) = \sum_{j=1}^{p} \log\{1 + \sum_{k=1}^{m} \psi_{\lambda_1}(d_{jk})\}$$

where ψ_λ is the SCAD penalty defined in (1). The objective function for our problem is formulated as

$$\min_{\mathbf{D}\in\mathcal{D}, \alpha_i \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{y}_i - \mathbf{D}\alpha_i||_2^2 + \Psi_{\lambda_1}(\mathbf{D}) + \lambda_2 \sum_{j=1}^{p} ||\alpha_j||_1. \tag{5}$$

Firstly, the GSCAD penalty tends to set small estimated d_{ij} to zero, and reduces the complexity of the estimated dictionary. If all elements in $\mathbf{d}_i$ are small, GSCAD will lead to $\mathbf{d}_i = 0$. Therefore, when starting with a relatively large p, GSCAD will be able to prune the dictionary by penalizing useless atoms to zero. In this way, the true size of the dictionary can be approximated by the number of non-zero columns in the resulting dictionary. In addition, if GSCAD detects some significant d_{ij}s in $\mathbf{d}_i$, it will exert less penalty on the whole $\mathbf{d}_i$ to avoid mistakenly truncating any real signals.

3.3 Algorithms

We follow the classic two steps approach to solve the optimization problem (5) iteratively. Given the dictionary $\mathbf{D}$, we update $\mathbf{A} = (\alpha_1, ..., \alpha)$ by solving the Lasso problem,

$$\min_{\alpha_i \in \mathbb{R}^p} \frac{1}{2} ||\mathbf{y}_i - \mathbf{D}\alpha_i||_2^2 + \lambda_2 ||\alpha_i||_1$$

for all signals $1 \leq i \leq n$. Given $\mathbf{A}$, the optimization problem (5) becomes

$$\arg\min_{\mathbf{D}\in\mathcal{C}} \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{y}_i - \mathbf{D}\alpha_i||_2^2 + \Psi_{\lambda_1}(\mathbf{D}), \tag{6}$$

which is addressed by the ADMM algorithm. Once $\mathbf{D}$ is updated, we remove all zero columns of $\mathbf{D}$ and reset p to the number of current atoms. Algorithm 1 demonstrates this whole procedure. It should be noted that (6) is a non-convex problem. Recently, the global convergence of ADMM in non-convex optimization is discussed in [19], which shows that several ADMM algorithms including SCAD are guaranteed to converge.

ADMM for updating dictionary. Problem (6) is equivalent to

$$\begin{aligned} &\min \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{y}_i - \mathbf{D_1}\alpha_i||_2^2 + \Psi_{\lambda_1}(\mathbf{D_2}) \\ &s.t. \quad \mathbf{D_1} = \mathbf{D_2}. \end{aligned}$$

We form the augmented Lagrangian as

$$L_\varrho(\mathbf{D_1}, \mathbf{D_2}, \xi) = \frac{1}{2} \sum_{i=1}^{n} ||\mathbf{y}_i - \mathbf{D_1}\alpha_i||_2^2 + \frac{\varrho}{2} ||\mathbf{D_1} - \mathbf{D_2}||_F^2 + \varrho ||\xi \circ (\mathbf{D_1} - \mathbf{D_2})||_F + \Psi_{\lambda_1}(\mathbf{D_2}).$$

where $\circ$ is the element-wise multiplication operator of two matrices, and $\xi \in \mathbb{R}^{d\times p}$. The ADMM algorithm consists three steps in each iteration

$$\mathbf{D_1}^{(t+1)} = \arg\min_{\mathbf{D_1}} L_\varrho(\mathbf{D_1}, \mathbf{D_2}^{(t)}, \xi^{(t)}) \tag{7}$$

$$\mathbf{D_2}^{(t+1)} = \arg\min_{\mathbf{D_2}} L_\varrho(\mathbf{D_1}^{(t+1)}, \mathbf{D_2}, \xi^{(t)}) \tag{8}$$

$$\xi^{(t+1)} = \xi^{(k)} + (\mathbf{D_1}^{(k+1)} - \mathbf{D_2}^{(k+1)}).$$

Problem (7) bears an explicit solution

$$\mathbf{D_1}^{(t+1)} \leftarrow \{\frac{1}{m}\mathbf{y}\mathbf{A}^T + \varrho(\mathbf{D_2}^{(t)} - \xi^{(t)})\}(\frac{1}{m}\mathbf{A}\mathbf{A}^T + \varrho I_r)^{-1}.$$

$\mathbf{D}_2$ in (8) can be solved by columnwise optimization such as

$$\mathbf{d_2}_j^{(t+1)} = \arg\min_{\mathbf{d_2}_j} \frac{\varrho}{2}||\mathbf{d_2}_j - (\mathbf{d_1}_j^{(t+1)} + \xi_j^{(t)}||_2^2 + \log\{1 + \Psi_{\lambda_1}(\mathbf{d_2}_j)\},$$

for $1 \le j \le p$. In theorem 1, we have shown that this is a convex problem under Condition (4), and can be solved easily by exiting convex optimization algorithms. The ADMM algorithm for updating dictionaries is summarized in Algorithm 2.

Algorithm 1: Dictionary Learning with GSCAD

Input : Training samples $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n]$, parameter λ_1,λ_2,c,m,p_0
initialize $\mathbf{D}^{(0)} \in \mathbb{R}^{m\times p_0}$ as random matrix with $d_{ij} \sim Unif(0, 1)$;
while *not converge* **do**
  Sparse Coding Stage: for $i = 1, ..., n$, update α_i by solving Lasso problem

$$\min_{\alpha_i\in\mathbb{R}^p} \frac{1}{2}||\mathbf{y}_i - \mathbf{D}\alpha_i||_2^2 + \lambda_2||\alpha_i||_1;$$

  Dictionary Update Stage: update $\mathbf{D}$ using Algorithm 2;
  Number of atoms: $p \leftarrow$ # columns of $\mathbf{D}$
end
Output : $\mathbf{D}$, p

Details in implementation. As demonstrated above, GSCAD has the ability to prune dictionary, and later in Section 4.1, we will see that its empirical performance is promising and competitive. However, if we initiate dictionary with a size smaller than the truth, there is nothing left for GSCAD to help. Therefore, an over-sized dictionary in the initiation step is strongly preferred. Experiments have shown that there is nothing to lose to start with a large dictionary as GSCAD can always prune it to the right size.

During the dictionary updating stage after we obtain a new dictionary from ADMM, if any two atoms are highly correlated, correlation greater than 0.95 for example, we only keep one of them. Some experiments have shown that this does not have much effect on the results, but will speed up convergence of the algorithm.

We define the convergence of the algorithm by the differences of $\mathbf{D}$ and the differences of $\mathbf{A}$ between two consecutive iterations. If they are both below a certain threshold, the algorithm stops. However, in implementation, we add an extra rule on the maximum number of iterations, since GSCAD may get stuck to a region where $\mathbf{D}$ keeps alternating from two local minima and never converge due to a bad initiation. Fortunately, the performance of local minima is mostly decent in terms of denoising.

Algorithm 2: Update dictionary using ADMM

Input : Training samples $\mathbf{Y}$, current $\mathbf{A} = (\alpha_1, ..., \alpha_n)$, parameter λ_1,c,ϱ

Initialize $\mathbf{D_2}^{(0)} = \xi = \mathbf{0} \in \mathbb{R}^{n \times p}$, set $t = 0$

while *not converge* **do**

$\mathbf{D_1}^{(t+1)} \leftarrow \{\mathbf{y}\mathbf{A}^T + \varrho(\mathbf{D_2}^{(t)} - \xi^{(t)})\}(\mathbf{A}\mathbf{A}^T + \varrho I_r)^{-1}$

Normalize each column of $\mathbf{D_1}$ as $\mathbf{d_1}_j \leftarrow \frac{1}{\max(||\mathbf{d_1}_j||_\infty, 1)} \mathbf{d_1}_j$;

Update $\mathbf{D_2}$: for $1 \leq j \leq p$,

$$\mathbf{d_2}_j^{(t+1)} = \arg\min_{\mathbf{d_2}_j} \frac{\varrho}{2}||\mathbf{d_2}_j - (\mathbf{d_1}_j^{(t+1)} + \xi_j^{(t)})||_2^2 + \log\{1 + \Psi_{\lambda_1}(\mathbf{d_2}_j)\}; \qquad (9)$$

$\xi^{(t+1)} \leftarrow \xi^{(k)} + (\mathbf{D_1}^{(k+1)} - \mathbf{D_2}^{(k+1)})$;

$t = t + 1$;

end

Remove the zero columns of $\mathbf{D_2}$;

Output : $\mathbf{D_2}$

4 Experimental Results

4.1 Synthetic Experiments

We design a simple example to check the performance of GSCAD from two aspects: (i) whether GSCAD could recover the true size of the dictionary, and (ii) its denoising performance compared with other methods.

Data generation. The generating dictionary $\mathbf{D}_0 \in \mathbb{R}^{10 \times 100}$ contains 10 atoms. Each atom is a vectoried 10×10 patch shown in Figure 2. Then 1500 signals $\{\mathbf{y}_i\}_{i=1}^{1500}$ in $\mathbb{R}^{100}$ are generated, each created by a linear combination of three different generating dictionary atoms picked randomly, with identically independently distributed coefficients following $Unif(0, 1/3)$. Gaussian noises $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are added, with signal-to-noise ratio (SNR) controlled by the Gaussian variance σ^2. Four levels of noise level are adopted at $\sigma = 0.01, 0.025, 0.05, 0.1$.

Applying GSCAD. In order to examine GSCAD's ability to prune dictionaries to the right size, dictionaries are initialized with varying number of atoms p_0, namely, 10 (true size), 15, 20 and 50. We run the GSCAD and received a learned dictionary $\hat{\mathbf{D}} \in \mathbb{R}^{m \times \hat{p}}$, where the resulting size of the dictionary $\hat{p}$ might be, and in most of the cases, is less than the initial size p_0. For validation, another 1000 signals are generated under the same setting. Both clean signals $\{\mathbf{x}_i^{test}\}_{i=1}^{1000}$ and noisy signals $\{\mathbf{y}_i^{test}\}_{i=1}^{1000}$ are recorded. Coefficients $\hat{\alpha}_i^{test} \in \mathbb{R}^{\hat{p}}$ corresponding to $\mathbf{y}_i^{test}$ are obtained using Orthogonal Matching Pursuit(OMP) with the number of non-zero elements fixed to three. We then reconstruct signals as $\hat{\mathbf{x}}_i^{test} = \hat{\mathbf{D}}\hat{\alpha}_i^{test}$, and calculate the PSNR as

$$\text{PSNR} = 10 \log_{10}(\frac{\sum_i ||\mathbf{x}_i^{test}||^2}{\sum_i ||\hat{\mathbf{x}}_i^{test} - \mathbf{x}_i^{test}||^2}).$$

Comparison. For each setting, we also run the K-SVD algorithm using the Matlab Toolbox associated its original paper Aharon et al. [1], and Online Learning algorithm [16] using the SPAMS package. Since neither K-SVD nor Online Learning would prune the dictionary, the learned dictionary $\hat{\mathbf{D}}$ will be in the same space as its initial value $\hat{\mathbf{D}}_0$, i.e. $\hat{p} = p_0$. Validation for both method are conducted in the same fashion as that in GSCAD.

Result. For each setting of σ and p_0, experiments using GSCAD, Online Learning and K-SVD are repeated 100 times. Median, first quartile and third quartile of the PSNR are

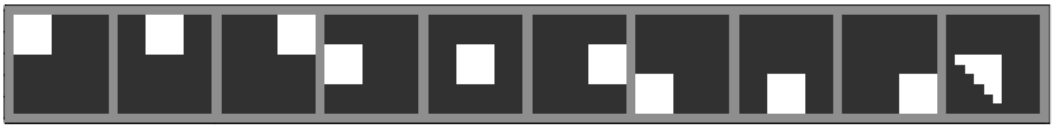

Figure 2: Atoms of the generating dictionary $\mathbf{D}_0$. Each atom corresponds to a 10×10 patch with white region representing 1 and black region representing 0.

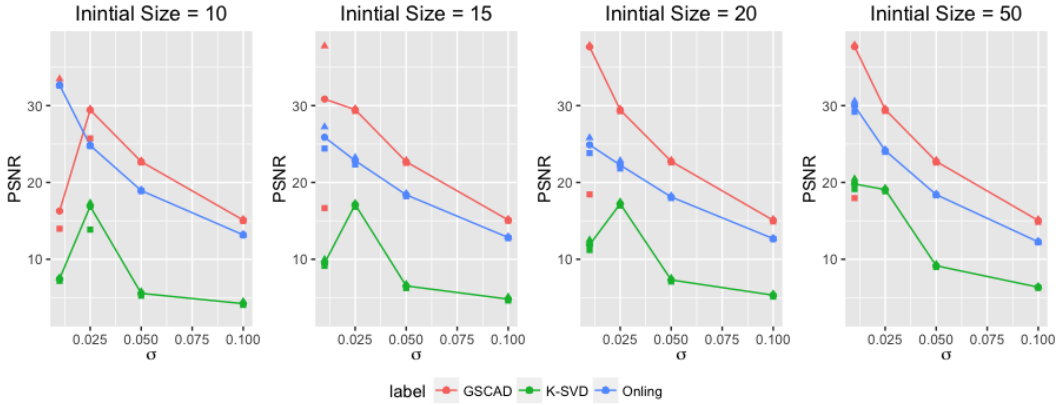

Figure 3: Synthetic results. Median PSNR: circle points connected by lines; first quartile: triangle-shaped points; third quartile: square-shaped points. Different algorithms are indexed by color.

shown in Figure 3. GSCAD performs better consistently than the other two methods when varying initial size p_0 and SNR levels controlled by σ, except just one case when initial p_0 is at the true value 10 and $\sigma = 0.01$. As suggested in Section 3.3, to make fully use of GSCAD's power of pruning, it is better to start with an over-sized initial dictionary. The mean and standard deviation of $\hat{p}$, the size of the resulting dictionary for GSCAD are reported in Table 1. The resulting size of the dictionary learned from GSCAD are very close to the truth, with very small standard deviations across all cases.

4.2 Image Denoising

Following the denoising scheme proposed by [6], we train our dictionaries directly on patches from corrupted images. More details about the scheme can be found in [12]. Five classical images (4) used in the image denoising benchmarks are corrupted with Gaussian noise. Standard deviations of Gaussian noise are set to be $\{5, 10, 20, 50\}$ separately, for pixel values in the range $[0, 255]$. For each corrupted image, overlapped 8×8 patches are obtained and centered as training set. For an image of size 512×512, a total number of 255025 patches $\mathbf{y}_i^c \in \mathbb{R}^{64}$ are extracted from the original image.

$p_0 \backslash \sigma$	0.01	0.025	0.05	0.1
10	9.97 (0.171)	10.00 (0)	10.00 (0)	10.00 (0)
15	10.02 (0.245)	10.03 (0.171)	10.15 (0.359)	10.14 (0.403)
20	10.02 (0.245)	10.08 (0.273)	10.15 (0.359)	10.35 (0.626)
50	10.07 (0.293)	10.07 (0.293)	10.19 (0.394)	10.36 (0.644)

Table 1: Average number of atoms in the resulting dictionary. Numbers in the parenthesis are corresponding standard deviations.

Figure 4: Images used in denoising: Barb, Boat, Fgprt, Lena, Peppers.

	Barb			Boat			Fgrpt		
σ	Gscad	Ksvd	Online	Gscad	Ksvd	Online	Gscad	Ksvd	Online
5	38.23	38.05	38.15	37.24	37.24	37.06	36.60	36.65	36.53
10	34.57	34.39	34.73	33.70	33.64	33.81	32.38	32.44	32.52
20	30.79	30.90	31.13	30.30	30.46	30.63	28.31	28.58	28.70
50	25.51	25.75	25.87	25.79	26.09	26.25	23.00	23.63	23.59
	Lena			Peppers			Average		
σ	Gscad	Ksvd	Online	Gscad	Ksvd	Online	Gscad	Ksvd	Online
5	38.63	38.60	38.56	38.04	37.78	37.81	37.75	37.66	37.62
10	35.58	35.48	35.70	34.51	34.22	34.71	34.15	34.03	34.29
20	32.31	32.41	32.60	30.84	30.82	31.22	30.51	30.63	30.86
50	27.66	27.88	28.02	25.84	26.32	26.37	25.56	25.93	26.02

Table 2: Average PSNR over five runs.

Dictionaries are trained using (i) the proposed GSCAD algorithm with $\lambda_1 = 0.05$ and $c = 3$ for Algorithm 2, (ii) K-SVD algorithm from the KSVD Matlab Toolbox, and (iii) Online Learning algorithm from the SPAMS package. Redundant DCT of size size $p = 256$ is used to initialize $\mathbf{D}$ for all three methods. The resulting dictionary is in $\mathbb{R}^{64\times\hat{p}}$, where $\hat{p}$ stays the same as p for K-SVD and Online Learning, but might be smaller for GSCAD.

Once a dictionary $\hat{\mathbf{D}}$ is obtained, patches $\mathbf{y}_i^c$ are approximated up to the noise level by a sparse linear combination of atoms in the dictionary:

$$\min_{\alpha_i\in\mathbb{R}^{\hat{p}}} ||\alpha_i||_0 \ s.t. \ ||\mathbf{y}_i^c - \hat{\mathbf{D}}\alpha_i||_2^2 \leq \epsilon_0,$$

where ϵ_0 is proportional to the noise variance σ^2. We set $\epsilon_0 = \sigma^2 F_m^{-1}(\tau)$ with $\tau = 0.9$ following the effective heuristic by [14]. F_m is the *cdf.* of χ^2 distribution with freedom of m. Then the denoised image is reconstructed based the sparse approximation $\hat{\mathbf{y}}_i^c = \hat{\mathbf{D}}\hat{\alpha}_i$, and its mean squared error (MSE) comparing to the clean image is calculated. For each setting, the whole procedure is repeated five times with different realizations of the noise. Define PSNR as

$$\text{PSNR} = 10\log_{10}(255^2/\text{MSE}).$$

The results for all three methods are very close to each other in general. At lower noise levels, GSCAD has a better performance. 3 summarizes the average number of atoms for the resulting dictionaries from GSCAD. It shows that GSCAD outperforms the other two methods with a learned dictionary less than half of the size of that used by the other algorithms. On the other hand, at higher noise levels, GSCAD becomes less competitive. Specifically, when $\sigma = 50$, all resulting $\hat{p}$ are close to the initial value of 256, which may indicate that dictionaries of size 256 is not large enough as an initial value for GSCAD. Our experience suggests that the higher noise level requires the larger dictionary size. Another interesting finding is that bearing the same level of noise, image 'fingerprint' needs a much larger dictionary to denoise compared with other images.

5 Conclusion

The GSCAD method has been presented for learning a sparse dictionary and selecting the dictionary size simultaneously. The experimental analysis has demonstrated very encouraging results relative to the state-of-the-art methods. This new framework may also be applied to the general subspace clustering problem for imaging clustering, which assumes

σ	Barb	Boat	Fgrpt	Lena	Peppers
5	109	125	178	86	100
10	92	94	195	80	86
20	120	151	224	136	153
50	248	248	246	250	247

Table 3: Number of atoms for the resulting dictionary ($\hat{p}$).

that similar points are described as points lying in the same subspace. The proposed formulation can learn the clustering and the number of clusters at the same time. This framework may also be applied to the architecture design of deep learning. The new GSCAD penalty can learn a sparse connection between units of two layers in the deep neural network to improve efficiency.

Appendix

Proof of Theorem 1.

(1). When $z_k = 0$, we have $(z_k - 0)^2 \leq (z_k - \theta_k)^2$, and further

$$\log\{1 + \psi_\lambda(0) + \sum_{l \neq k} \psi_\lambda(\theta_l)\} \leq \log\{1 + \psi_\lambda(\theta_k) + \sum_{l \neq k} \psi_\lambda(\theta_l)\},$$

for any $\theta_k \in \mathbb{R}$. When $z_k \neq 0$, we have

$$\{z_k - sign(z_k)|\theta_k|\}^2 \leq [z_k - \{-sign(z_k)|\theta_k|\}]^2,$$

and further

$$\log\{1 + \psi_\lambda(sign(z_k)|\theta_k|) + \sum_{l \neq k} \psi_\lambda(\theta_l)\} = \log\{1 + \psi_\lambda(-sign(z_k)|\theta_k|) + \sum_{l \neq k} \psi_\lambda(\theta_l)\}.$$

Therefore to minimize(2), $\hat{\theta}_k$ has to satisfy $sign(\hat{\theta}_k) = sign(z_k)$. If we denote $\tilde{K} = \{1 \leq k \leq K : \ z_k \neq 0\}$ and denote Θ_k as the open interval between z_k and 0, i.e.

$$\Theta_k = \begin{cases} (0, z_k), & if \ z_k > 0 \\ (z_k, 0), & if \ z_k < 0 \end{cases},$$

then optimization problem (2) is equivalent to

$$\min_{\theta_k \in \Theta_k \cup \{0\}, k \in \tilde{K}} \frac{\varrho}{2} \sum_{k \in \tilde{K}} (z_k - \theta_k)^2 + \log\{1 + \sum_{k \in \tilde{K}} \psi_\lambda(\theta_k)\}.$$

(2). Recall that $c_0 = card(\tilde{K})$. To simplify the notation, we rewrite $z = (z_1, ..., z_{c_0}) \in \mathbb{R}^{c_0}$ as if there were no zero-element in z, and correspondingly $\theta = (\theta_1, ..., \theta_{c_0}) \in \mathbb{R}^{c_0}$. Define $L : \mathbb{R}^{c_0} \to \mathbb{R}$:

$$L(\theta) = \frac{\varrho}{2}||z_k - \theta_k||^2 + \log\{1 + \sum_{k=1}^{c_0} \psi_\lambda(\theta_k)\}.$$

We expend Θ_k to the whole half plane:

$$\tilde{\Theta}_k = \begin{cases} (0, \infty), & if \ z_k > 0 \\ (-\infty, 0), & if \ z_k < 0 \end{cases}.$$

If we can show that L is convex in $\tilde{\Theta}_1 \times ... \times \tilde{\Theta}_{c_0}$, this will inply that L is convex over $\prod_{k=1}^{c_0} \Theta_k \cup \{0\}$, as L is continous all over $\mathbb{R}^{c_0}$.

To show that the optimization problem within $\Theta^o = \tilde{\Theta}_1 \times ... \times \tilde{\Theta}_{c_0}$ is convex, we are going to varify the inequality

$$L((1-t)x + ty) \leq (1-t)L(x) + tL(y), \quad t \in [0,1],$$

for any $x, y \in \Theta^o$. This is trivial for $x = y$, and for $x \neq y$, we consider the following two cases.

Case 1: $x, y \in \Theta_1^o = \{x \in \Theta^o : |x_i| \notin \{\lambda, c\lambda\} \text{ for any } 1 \leq i \leq c_0\}$. Therefore only a finite number of points in set $\{tx + (1-t)y : t \in [0,1]\}$ such that L does not have a second-order derivative. Let $v = x - y$. Define $\varphi(t) = L(x + tv), t \in [0,1]$. If we can show that $\varphi'(t)$ is continuous on $[0,1]$, and $\varphi''(t) \geq 0$ except at a finite number of points, therefore $\varphi'(t)$ is non-decreasing, and furthermore $\varphi(t)$ is convex on $[0,1]$. By definition, for any $t \in [0,1]$,

$$L((1-t)x + ty) = L(x + tv) = \varphi(t) \leq t\varphi(1) + (1-t)\varphi(0) = tL(y) + (1-t)L(x).$$

Therefore L is convex.

Now we are going to show that $\varphi'(t)$ is continuous and $\varphi''(t) \geq 0$ except at a finite number of points, where $\varphi''(t)$ does not exist. Taking derivative of L, we get

$$\begin{aligned}
L'_{x_i} &= sign(x_i)\{\varrho|x_i| + \frac{\dot{\psi}_\lambda(x_i)}{1 + \sum_k \psi_\lambda(x_k)}\} - \varrho z_k, \\
L''_{x_i x_i} &= \varrho + \frac{\ddot{\psi}_\lambda(x_i)}{1 + \sum_k \psi_\lambda(x_k)} - \frac{\dot{\psi}^2_\lambda(x_i)}{\{1 + \sum_k \psi_\lambda(x_k)\}^2}, \quad |x_i| \notin \{\lambda, c\lambda\}, \\
L''_{x_i x_j} &= -\frac{\dot{\psi}_\lambda(x_i) \cdot \dot{\psi}_\lambda(x_j)}{\{1 + \sum_k \psi_\lambda(x_k)\}^2}, \quad |x_i|, |x_j| \notin \{\lambda, c\lambda\}
\end{aligned}$$

where

$$\dot{\psi}_\lambda(x_i) = \begin{cases} \lambda \cdot sign(x_i), & if\ |x_i| \leq \lambda \\ \frac{c\lambda - |x_i|}{(c-1)} \cdot sign(x_i), & if\ \lambda < |x_i| \leq c\lambda \\ 0, & if\ |x_i| > c\lambda \end{cases} \quad and \quad \ddot{\psi}_\lambda(x_i) = \begin{cases} -\frac{1}{(c-1)}, & if\ \lambda < |x_i| \leq c\lambda \\ 0, & o.w. \end{cases}.$$

Since L'_{x_i} is continuous for all $1 \leq i \leq c_0$ and $x \in \Theta^o$,

$$\varphi'(t) = \sum_i \frac{\partial L}{\partial x_i}(x + tv) \cdot v_i$$

is contious. Except a finite number of $t \in [0,1]$, such that $L''_{x_i x_j}$ does not exist at $x + tv$, we have

$$\begin{aligned}
\varphi''(t) &= \sum_{i,j} \frac{\partial^2 L}{\partial x_i \partial x_j}(x + tv) v_i v_j \\
&= \sum_{i=1}^{c_0} \{\varrho + \frac{\ddot{\psi}_\lambda(x_i)}{1 + \sum_k \psi_\lambda(x_k)}\} v_i^2 - \{1 + \sum_k \psi_\lambda(x_k)\}^{-2} \{\sum_{i=1}^{c_0} \dot{\psi}_\lambda(x_i) v_i\}^2 \\
&\geq \sum_{i=1}^{c_0} \{\varrho + \frac{\ddot{\psi}_\lambda(x_i)}{1 + \sum_k \psi_\lambda(x_k)}\} v_i^2 - \{1 + \sum_k \psi_\lambda(x_k)\}^{-2} c_0 \sum_{i=1}^{c_0} \dot{\psi}^2_\lambda(x_i) v_i^2 \\
&= \sum_{i=1}^{c_0} \{\varrho + \frac{\ddot{\psi}_\lambda(x_i)}{1 + \sum_k \psi_\lambda(x_k)} - \frac{c_0 \dot{\psi}^2_\lambda(x_i)}{\{1 + \sum_k \psi_\lambda(x_k)\}^2}\} v_i^2.
\end{aligned}$$

Let

$$f_i(x_i) = \varrho + \frac{\ddot{\psi}_\lambda(x_i)}{1+\sum_l \psi_\lambda(b_l)} - \frac{c_0\,\dot{\psi}^2_\lambda(x_i)}{\{1+\sum_l \psi_\lambda(b_l)\}^2}, \quad 1 \le i \le c_0.$$

To show that $\varphi''(t) \ge 0$, we only need to show that $f_i(x_i) \ge 0$. Since $f_i(x_i) = f_i(-x_i)$, without loss of generality, we are only going to show that $f_i(x_i) \ge 0$, for $x_i > 0$.

Take derivative of f_i,

$$f_i'(x_i) = -\frac{\dddot{\psi}_\lambda(x_i)\dot{\psi}_\lambda(x_i)}{1+\sum_l \psi_\lambda(x_l)} - \frac{2c_0\,\dot{\psi}^2_\lambda(x_i)\ddot{\psi}_\lambda(x_i)}{\{1+\sum_l \psi_\lambda(x_l)\}^2} + \frac{2c_0\,\dot{\psi}^3_\lambda(x_i)}{\{1+\sum_l \psi_\lambda(x_l)\}^3}, \quad x_i \notin \{\lambda, c\lambda\}.$$

Since $\ddot{\psi}_\lambda(x_i) \le 0$ and $\dot{\psi}_\lambda(x_i) \ge 0$, we have $f_i'(x_i) \ge 0$ for all $x_i \in \tilde{\Theta}_k \backslash \{\lambda, c\lambda\}$. Observe that $f_i(x_i)$ is piece-wise continous on $(0,\lambda)$, $(\lambda, c\lambda)$, and $(c\lambda, \infty)$. For $x_i \in (0,\lambda)$,

$$f_i(x_i) \ge \lim_{x_i \to 0^+} f_i(x_i) = \varrho - \frac{c_0\lambda^2}{\{1+\sum_{l\in\tilde{K}, l\neq k} p_\lambda(x_l)\}^2} \ge \varrho - c_0\lambda^2 \ge 0.$$

For $x_i \in (\lambda, c\lambda)$

$$\begin{aligned}
f_i(x_i) &\ge \lim_{x_i \to \lambda^+} f_i(x_i) \\
&= \varrho - \frac{1}{(c-1)\{1+\lambda^2+\sum_{l\neq k}\psi_\lambda(x_l)\}} - \frac{c_0\lambda^2}{\{1+\lambda^2+\sum_{l\neq k}\psi_\lambda(x_l)\}^2} \\
&\ge \varrho - \frac{1}{(c-1)(1+\lambda^2)} - \frac{c_0\lambda^2}{(1+\lambda^2)^2} \\
&= \frac{\varrho(c-1)(1+\lambda^2)^2 - (1+\lambda^2) - c_0(c-1)\lambda^2}{(c-1)(1+\lambda^2)^2} \\
&\ge 0.
\end{aligned}$$

For $x_i \in (c\lambda, \infty)$,

$$f_i(x_i) \ge \lim_{x_i \to c\lambda^+} f_i(x_i) = \varrho > 0.$$

Therefore $f_i(x_i) \ge 0$, for $x_i > 0$, and furthermore, $\varphi''(t) \ge 0$ except a finite number of $t \in [0,1]$. Thus we finished the proof of case 1.

Case 2: $x \in \Theta_0^o$ or $y \in \Theta_0^o$, where $\Theta_0^o = \Theta^o \backslash \Theta_1^o = \{x \in \Theta^o : |x_i| = \lambda, \text{or } c\lambda, \text{ for some } 1 \le i \le c_0\}$. Without loss of generality, we assume that the last $c_0 - k$, $1 \le k \le n$ elements of x and y are the same, and the rest are not, i.e. $x_i \neq y_i$ for $1 \le i \le k$ and $x_i = y_i$ for $k+1 \le i \le c_0$. Let $x^* = (x_1, ..., x_k)$, $y^* = (y_1, ..., y_k)$ and $v^* = y^* - x^*$. Therefore only a finite number of $t \in [0,1]$ such that point $(1-t)x^* + ty^*$ belongs to $\mathcal{D}^k = \{x \in \tilde{\Theta}_1 \times ... \times \tilde{\Theta}_k : |x_i| = \lambda, \text{or } c\lambda, \text{ for some } 1 \le i \le k\}$.

Let $w = (w_1, ..., w_k)$, and define $g : \tilde{\Theta}_{i_1} \times ... \times \tilde{\Theta}_{i_k} \to \mathbb{R}$, as

$$g(w) = L((w, x_{k+1}, ..., x_{c_0})).$$

Define $\varphi^*(t) = g(x^* + tv^*), t \in [0,1]$. Then similar to Case 1, we can show that

$$\frac{d\varphi^*}{dt} = \sum_i \frac{\partial g}{\partial x_i^*}(x^* + tv^*) \cdot v_i^* = \sum_{i=1}^k \frac{\partial L}{\partial x_i}\big((x^* + tv^*, x_{k+1}, ..., x_n)\big) \cdot v_i^*$$

is contious, and

$$\begin{aligned}\frac{d^2\varphi^*}{dt^2} &= \sum_{i,j} \frac{\partial^2 g}{\partial x_i^* \partial x_j^*}(x^* + tv^*)v_i^* v_j^* \\ &= \sum_{i,j=1}^{k} \frac{\partial^2 L}{\partial x_i \partial x_j}\big((x^* + tv^*, x_{k+1}, ..., x_n)\big)v_i^* v_j^* \\ &\geq 0\end{aligned}$$

except a finite number of $t \in [0,1]$. Therefore $d\varphi^*/dt$ is non-decreasing, and further $\varphi^*(t)$ is convex on $[0,1]$. By definition, for any $t \in [0,1]$,

$$\begin{aligned}L((1-t)x + ty) = L(x + tv) &= g(x^* + tv^*) \\ &= \varphi^*(t) \leq t\varphi^*(1) + (1-t)\varphi^*(0) = tL(y) + (1-t)L(x).\end{aligned}$$

Thus, we finished the proof of case 2.

References

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

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[3] O. Bryt and M. Elad. Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008.

[4] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8): 1207–1223, 2006.

[5] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4): 1289–1306, 2006.

[6] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[7] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[8] E. Gassiat and R. Van Handel. Consistent order estimation and minimal penalties. *Information Theory, IEEE Transactions on*, 59(2):1115–1128, 2013.

[9] Z. Geng, S. Wang, M. Yu, P. O. Monahan, V. Champion, and G. Wahba. Group variable selection via convex log-exp-sum penalty with application to a breast cancer survivor study. *Biometrics*, 71(1):53–62, 2015.

[10] J. Huang, S. Ma, H. Xie, and C.-H. Zhang. A group bridge approach for variable selection. *Biometrika*, 96(2):339–355, 2009.

[11] S. Kong and D. Wang. A dictionary learning approach for classification: separating the particularity and the commonality. In *Computer Vision–ECCV 2012*, pages 186–199. Springer, 2012.

[12] J. Mairal and F. Bach. Sparse Modeling for Image and Vision Processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2):85–283, 2014. ISSN 1572-2740.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[15] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

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

[17] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791–804, 2012.

[18] M. Ranzato, C. Poultney, S. Chopra, Y. L. Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2006.

[19] Y. Wang, W. Yin, and J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*, 2015.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[21] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801, 2009.

[23] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[24] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.

[25] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Department of Statistics, UC Berkeley, Tech. Rep*, 703, 2006.

[26] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in neural information processing systems*, pages 2295–2303, 2009.

[27] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, 21(1):130–144, 2012.

[28] N. Zhou and J. Zhu. Group variable selection via a hierarchical lasso and its oracle property. *arXiv preprint arXiv:1006.2871*, 2010.