ACCURATE DICTIONARY LEARNING WITH DIRECT SPARSITY CONTROL

Hongyu Mou, Adrian Barbu

Statistics Department, Florida State University
Tallahassee FL 32306

ABSTRACT

Dictionary learning is a popular method for obtaining sparse linear representations for high dimensional data, with many applications in image classification, signal processing and machine learning. In this paper, we introduce a novel dictionary learning method based on a recent variable selection algorithm called Feature Selection with Annealing (FSA). Because FSA uses an L_0 constraint instead of the L_1 penalty, it does not introduce any bias in the coefficients and obtains a more accurate sparse representation. Furthermore, the L_0 constraint makes it easy to directly specify the desired sparsity level instead of indirectly through a L_1 penalty. Finally, experimental validation on real gray-scale images shows that the proposed method obtains higher accuracy and efficiency in dictionary learning compared to classical methods based on the L_1 penalty.

Index Terms— sparse coding, dictionary learning, FSA, LARS

1. INTRODUCTION

Sparse dictionary learning is a feature learning method that aims to represent the input data as a linear combination of a small number of elements of a dictionary, called atoms. Unlike principal component analysis, the atoms in the dictionary are not required to be orthogonal. Furthermore, the dictionary usually contains more atoms that the dimensionality of the data, which means that the dictionary gives an over-complete representation.

In 1997, Olshausen [1] introduced the idea of sparse coding with an overcomplete basis, as a possible explanation of the receptive cells in the V1 part of the brain. There has been a lot of work on sparse coding since then, which could be grouped into different categories based on the type of regularization that was used to obtain the sparse representation. A good survey of the different methods has been done in [2].

Given an observation $\mathbf{x} \in \mathbb{R}^d$ and a dictionary $\mathbf{D}$ with p atoms as a $d \times p$ matrix, sparse coding can be obtained by minimizing the l_0-norm with the constraint of exact reconstruction [3]:

$$\boldsymbol{\alpha} = \operatorname{argmin} ||\boldsymbol{\alpha}||_0 \text{ s.t. } \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$$

where $||\boldsymbol{\alpha}||_0$ is the number of nonzero elements in the sparse vector $\boldsymbol{\alpha}$.

Because l_0-norm minimization is an NP-hard problem, a popular approximation used in machine learning and statistics [4, 5] is the l_1-norm minimization, especially the Lasso [6]:

$$\boldsymbol{\alpha} = \operatorname*{argmin}_{\boldsymbol{\alpha}} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \quad (1)$$

where the penalty parameter λ can be tuned for a desired sparsity of the solution. One can also obtain the entire regularization path using least angle regression(LARS)[7]. Recently, other more efficient methods for global optimization of the convex loss (1) have been proposed using gradient projection and homotopy [8].

Another way to introduce sparse coding is through l_p-norm minimization, where $0 < p < 1$. There are three typical algorithms for l_p minimization [9]: General Iteratively Reweighted Least Squares (GIRLS), Iteratively Thresholding Method (ITM) and Iteratively Reweighted Least Squares (IRLS). Experimental comparison of the three methods revealed that IRLS has the best performance and is the fastest as well. Furthermore, l_p minimization with IRLS has been used for robust face recognition [10].

Most recent methods for dictionary learning are based on regression with sparsity inducing penalties such as the convex l_1 penalty or nonconvex penalties such as the SCAD penalty [11]. Examples include the online dictionary learning [12] and Fisher discrimination dictionary learning [13]. Dictionary learning has been applied to face recognition using discriminative K-SVD [14].

In this paper we will investigate a novel approach to sparse dictionary learning based on a recent l_0-based optimization method named Feature Selection with Annealing (FSA)[15]. In our context of dictionary learning, FSA can be used for solving the l_0-constraint optimization problem:

$$\boldsymbol{\alpha} = \operatorname*{argmin}_{||\boldsymbol{\alpha}||_0 \le k} ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2$$

where k is the desired number of nonzero entries of $\boldsymbol{\alpha}$. Compared with Lasso, the FSA method can directly control the sparsity of the solution and has better performance than l_1-norm and l_p-norm minimization methods.

Through experiments we will see that FSA obtains a more accurate reconstruction of the data for the same sparsity level than LARS or ILRS, while being faster. We will also see

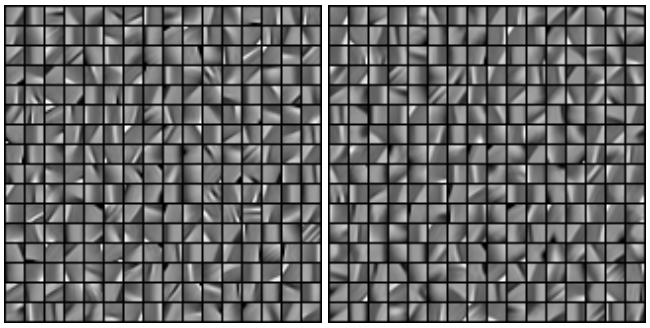

Fig. 1. Learned dictionaries. Left: dictionary obtained using the l_1 penalty. Right: dictionary obtained using FSA

that the obtained dictionary can be used for classification on the MNIST data using different learning methods, obtaining better results than LARS, IRLS and direct learning without sparse coding.

2. DICTIONARY LEARNING WITH FSA

According to the classical dictionary learning [1], suppose we have an input data set $\mathbf{X} \in \mathbb{R}^{d\times n}$, each column representing an $m \times m$ image patch and $n >> d$ is the number of extracted image patches. The goal of dictionary learning is to find a dictionary $\mathbf{D} \in \mathbb{R}^{d\times p}$ where each column is an atom and:

$$\mathbf{X} \approx \mathbf{DA}$$

where $\mathbf{A} = (\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_n)$ contains sparse vectors $\boldsymbol{\alpha}_i \in \mathbb{R}^p$.

Learning the dictionary $\mathbf{D}$ and the sparse representation $\mathbf{A}$ is done by minimizing a cost function:

$$f_n(\mathbf{D}, \mathbf{A}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\alpha}_i) \tag{2}$$

where the loss function $l(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\alpha}_i)$ measures the difference between $\mathbf{x}_i$ and $\mathbf{D}\boldsymbol{\alpha}_i$ and encourages a sparse $\boldsymbol{\alpha}_i$.

One popular loss function is the l_1-penalized square loss also known as the Lasso [6, 16, 7]:

$$l(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\alpha}_i) \triangleq \frac{1}{2}||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1 \tag{3}$$

where λ is a regularization parameter that imposes the desired level of sparsity for $\boldsymbol{\alpha}_i$. The Lasso has been used in sparse sparse dictionary learning before, for example in online dictionary learning [12].

However, the l_1 approach has the disadvantage that it controls the sparsity of $\boldsymbol{\alpha}_i$ only indirectly through the penalty λ. We are interested in an approach where the sparsity of $\boldsymbol{\alpha}_i$ can be directly specified as a constraint $||\boldsymbol{\alpha}_i||_0 \leq k$.

$$l(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\alpha}_i) \triangleq \begin{cases} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 & \text{if } ||\boldsymbol{\alpha}_i||_0 \leq k \\ \infty & \text{else} \end{cases} \tag{4}$$

The cost function (2) depends on two variables: $\mathbf{D}$ and $\mathbf{A}$. We will use a straightforward approach that minimizes the cost (2) by alternately minimizing over one variable while keeping the other one fixed. When $\mathbf{D}$ is fixed, minimizing over $\mathbf{A}$ can be obtained by independently minimizing (4) for each $\mathbf{x}_i$.

The whole alternating procedure for dictionary learning is described in Algorithm 1. It depends on two other algorithms, which will be described in the next two sections.

Algorithm 1 Dictionary learning with FSA

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ with n observations $\mathbf{x}_1, ..., \mathbf{x}_n$ as columns, sparsity level k.
Output: Trained dictionary $\mathbf{D}$.

1: Initialize $\mathbf{D}_0$ with p random observations from $\mathbf{X}$.
2: **for** t=1 to T **do**
3: **for** i=1 to n **do**
4: Use Algorithm 2 to compute
$$\boldsymbol{\alpha}_i = \underset{||\boldsymbol{\alpha}||_0 \leq k}{\operatorname{argmin}} ||\mathbf{x}_i - \mathbf{D}_{t-1}\boldsymbol{\alpha}||_2^2$$
5: **end for**
6: Set $\mathbf{A} = (\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_n) \in \mathbb{R}^{p\times n}$.
7: Compute $\mathbf{D}_t$ by Algorithm 3, with $(\mathbf{X}, \mathbf{A}, \mathbf{D}_{t-1})$ as input
$$\mathbf{D}_t = \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{D}_{t-1}\boldsymbol{\alpha}_i||_2^2 \tag{5}$$
8: **end for**
9: Return $\mathbf{D} = \mathbf{D}_T$

2.1. Feature Selection with Annealing (FSA)

FSA is a novel variable selection method introduced in [15] that minimizes a differentiable loss function $L(\boldsymbol{\alpha})$ with sparsity constraints

$$\boldsymbol{\alpha} = \underset{||\boldsymbol{\alpha}||_0 \leq k}{\operatorname{argmin}} L(\boldsymbol{\alpha}).$$

In our case, the loss function $L(\boldsymbol{\alpha})$ is:

$$L(\boldsymbol{\alpha}) = ||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2$$

where $\mathbf{x}$ is any of the columns of $\mathbf{X}$ and $\mathbf{D}$ is the current dictionary.

FSA achieves sparsity by gradually reducing the dimensionality of $\boldsymbol{\alpha}$ from p to k, according to an annealing schedule. It starts with a full $\boldsymbol{\alpha} \in \mathbb{R}^p$ and alternates the removal of some variables with the updating of the remaining parameters by gradient descent. The whole procedure is described in Algorithm 2.

Algorithm 2 Feature Selection with Annealing

Input: Observation $\mathbf{x} \in \mathbb{R}^d$, current dictionary $\mathbf{D} \in \mathbb{R}^{d\times p}$, sparsity level k
Output: Sparse $\boldsymbol{\alpha} \in \mathbb{R}^p$ with $\|\boldsymbol{\alpha}\|_0 \leq k$.

1: Initialize $\boldsymbol{\beta} = 0 \in \mathbb{R}^p, J = \{1, ..., p\}$
2: **for** e=1 to N^{iter} **do**
3: Update $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \eta\mathbf{D}^T(\mathbf{D}\boldsymbol{\beta} - \mathbf{x})$
4: Find the indexes $I, |I| = p_e$ corresponding to the highest p_e elements of $|\boldsymbol{\beta}|$.
5: Keep only the entries with index I in $\boldsymbol{\beta}, \mathbf{x}, J$ and $\mathbf{D}$, i.e.
$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}_I, \mathbf{x} \leftarrow \mathbf{x}_I, J \leftarrow J_I, \mathbf{D} \leftarrow \mathbf{D}_{II}$$
6: **end for**
7: Set $\boldsymbol{\alpha} = 0 \in \mathbb{R}^p$, then $\boldsymbol{\alpha}_J = \boldsymbol{\beta}$.

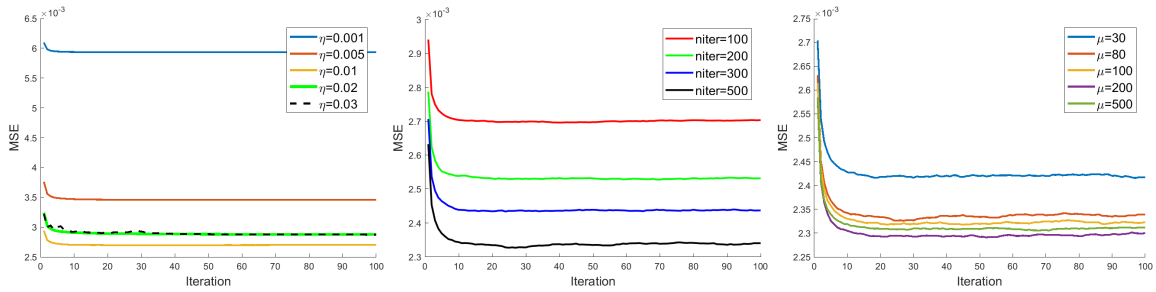

Fig. 2. Data reconstruction error vs number of learning iterations for the three FSA hyper-parameters: η, N^{iter} and μ.

The annealing schedule p_e represents the number of non-zero variables that are kept at the e^{th} iteration. A fast annealing schedule will save computation time but loose some accuracy in selecting the correct variables. So it is important to find a proper annealing schedule that balances speed and accuracy. In [15], the authors provide an inverse schedule:

$$p_e = k + (p-k)\max(0, \frac{N^{iter} - 2e}{2e\mu + N^{iter}})$$

where p is the number of total variables and k is the sparsity level. The parameter μ controls the speed of removing the variables. Together with the learning rate η, they can be tuned to obtain a small value of the loss function at the completion of the algorithm.

2.2. Dictionary Update

The loss function (5) is quadratic in $\mathbf{D}$ and could be minimized analytically. To avoid large matrix operations, we use block-coordinate descent with warms starts, as described in [12] and in Algorithm 3 below.

Algorithm 3 Dictionary Update

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, sparse matrix $\mathbf{A} = (\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_n) \in \mathbb{R}^{p\times n}$, input dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k] \in \mathbb{R}^{d\times p}$

Output: dictionary $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k] \in \mathbb{R}^{d\times p}$

Set $\mathbf{B} = \mathbf{A}\mathbf{A}^T \in \mathbb{R}^{p\times p}$,
Set $\mathbf{C} = \mathbf{X}\mathbf{A}^T = [\mathbf{c}_1, ..., \mathbf{c}_p] \in \mathbb{R}^{d\times p}$.
repeat
  for $j = 1$ to p **do**
    Set $\mathbf{u}_j \leftarrow \frac{1}{\mathbf{B}_{jj}}(\mathbf{c}_j - \mathbf{D}\mathbf{b}_j) + \mathbf{d}_j$
    Set $\mathbf{d}_j \leftarrow \frac{\mathbf{u}_j}{\max(||\mathbf{u}_j||_2, 1)}$
  end for
until convergence
Return updated dictionary $\mathbf{D}$.

3. EXPERIMENTS

Data Description. In the experiments we use two standard gray images: Lena and Boat, resized to 128×128. We work with overlapping patches of size of 9×9 extracted from these images, and our input data is $\mathbf{X} \in \mathbb{R}^{81\times 14400}$.

For evaluating the quality of the learned dictionary $\mathbf{D}$ we will use the MSE of the data reconstruction:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2$$

FSA Parameter Experiments. First, we need to find proper hyper-parameter values for μ, η, and N^{iter} for using FSA in dictionary learning. We experimented with learning a 1024 atom dictionary on the Lena image. In Figure 2, left are shown the MSE vs iteration number for different values the learning rate η, for $N^{iter} = 100$ and $\mu = 80$. We see that the quality of the dictionary is almost the same for η between 0.01 and 0.03. In Figure 2, middle are shown the MSE vs iteration number for different values of N^{iter}, when $\mu = 80, \eta = 0.01$. We see that with more iterations, the quality of the dictionary is better, at an increased computation cost. We fixed $N^{iter} = 500$, which has a comparable computation cost with LARS. In Figure 2, right are shown the MSE vs iteration number for different values the annealing parameter μ, for $N^{iter} = 500$ and $\eta = 0.01$. We see that the quality of the dictionary is almost the same for μ between 100 and 500, the best being for $\mu = 200$.

Comparison with other methods. In this experiment we compare the dictionary learned with FSA with approaches where Algorithm 2 was replaced by LARS or IRLS. We investigated three sparsity levels $k = 3, 4, 5$.

In Figure 3 are shown the reconstruction MSE for the three methods for 256 and 1024 atoms on Lena, and 1024 atoms on the Lena and Boats images simultaneously. We see that FSA obtains better dictionaries than LARS which is better than IRLS. Furthermore, FSA with $k = 4$ has smaller MSE than LARS and IRLS with $k = 5$ in all three cases. Moreover, IRLS is at least 10 times slower than FSA, which is why we didn't include the Lena+Boats results for IRLS.

Examples of reconstructed images with the learned dictionary are shown in Figure 4. We can see that FSA obtains a more accurate and more clear image than LARS.

Digit Recognition Application. In this section, we use dictionary learning to obtain a sparse representation of the MNIST [17] handwritten digit database and apply several multi-class classification methods on the sparse feature vectors to compare the classification accuracy.

MNIST is a dataset containing 60,000 training examples and 10,000 test examples as grayscale images of size 28×28

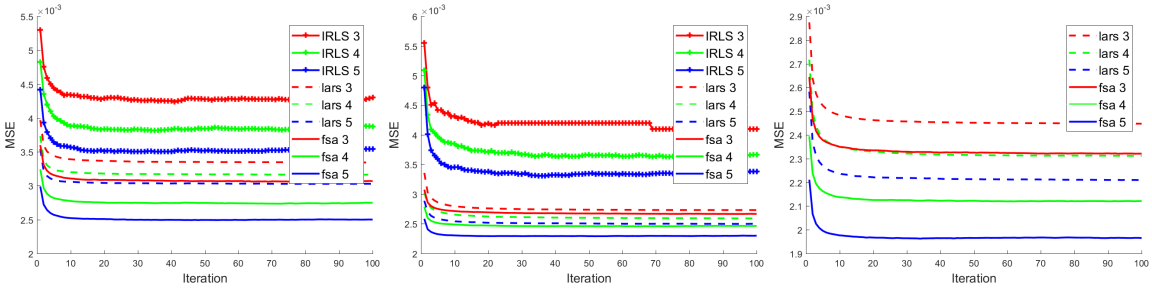

Fig. 3. Comparison with other methods. Left: dictionary with 256 atoms on Lena. Middle: dictionary with 1024 atoms on Lena. Right: dictionary with 1024 atoms on Lena+Boats.

of handwritten digits.

Fig. 4. Reconstructed images. Top: original images. Middle: images reconstructed using LARS. $MSE_{Lena} = 0.0027, MSE_{Boat} = 0.0035$. Bottom: images reconstructed using FSA. $MSE_{Lena} = 0.0024, MSE_{Boat} = 0.0031$.

In our experiment, we generate 256-atom dictionaries using different dictionary learning methods, one dictionary for each digit from 0 to 9. Then for each observation we generated one sparse feature vector from each dictionary, and concatenated them to obtain a 2560 dimensional sparse feature vector. This sparse feature vector was used as input for training and testing different classifiers.

As classifiers, we used SVM, Random Forest with 500 trees, and K-Nearest Neighbors with $K = 3$. We compared

Table 1. Test misclassification errors on MNIST data.

Method	Sparsity	SVM	KNN(K=3)	RF
FSA	5	0.0362	0.0915	0.0915
FSA	10	0.0257	0.0682	0.0394
FSA	20	**0.0223**	0.0561	0.0331
FSA	50	0.0239	**0.0364**	**0.0270**
LARS	5	0.0372	0.1062	0.0564
LARS	10	0.0323	0.1312	0.0463
LARS	20	0.0300	0.1012	0.0397
LARS	50	0.0330	0.0802	0.0357
IRLS	5	0.0848	0.1585	0.1030
IRLS	10	0.0759	0.1501	0.0904
IRLS	20	0.0527	0.1465	0.0582
IRLS	50	0.0503	0.0955	0.0539
Original data	-	0.0562	0.0619	0.0352

different sparsity levels in the sparse representation, and also tested the classifiers on the original data with no sparsity.

The results are shown in Table 1. We see that for each type of classifier the sparse representation obtained by FSA has smaller test misclassification error than LARS, IRLS and the original data.

4. CONCLUSION

We introduced a new method to solve the sparse coding problem in dictionary learning that replaces the L_1 penalty in the loss function with a sparsity constraint $||\boldsymbol{\alpha}||_0 \leq k$. The method relies on a recent feature selection methods called Feature Selection with Annealing (FSA). Using FSA we can directly specify the number of non-zero variables we want in the sparse representation, unlike the L_1 penalized methods where the sparsity is controlled indirectly through the regularization parameter λ.

The experimental results on image reconstruction showed that the proposed method obtains smaller reconstruction errors than LARS or IRLS for the same sparsity level and dictionary size. Furthermore, experiments on the MNIST dataset using SVM, Random Forest and K-Nearest Neighbors showed that the method can be used to obtain a sparse image representation that obtains a smaller misclassification error on than directly using the image as input, Furthermore the sparse representation by FSA again outperforms LARS and IRLS in this classification task.

5. 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] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang, "A survey of sparse representation: algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[3] David L Donoho and Michael Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[4] Vishal M Patel and Rama Chellappa, "Sparse representations, compressive sensing and dictionaries for pattern recognition," in *ACPR*, 2011, pp. 325–329.

[5] Yong Yuan, Xiaoqiang Lu, and Xuelong Li, "Learning hash functions using sparse reconstruction," in *Proceedings of International Conference on Internet Multimedia Computing and Service*, 2014, p. 14.

[6] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[7] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[8] Allen Y Yang, S Shankar Sastry, Arvind Ganesh, and Yi Ma, "Fast l1-minimization algorithms and an application in robust face recognition: A review," in *ICIP*, 2010, pp. 1849–1852.

[9] Qin Lyu, Zhouchen Lin, Yiyuan She, and Chao Zhang, "A comparison of typical lp minimization algorithms," *Neurocomputing*, vol. 119, pp. 413–424, 2013.

[10] Song Guo, Zhan Wang, and Qiuqi Ruan, "Enhancing sparsity via lp (0¡ p¡ 1) minimization for robust face recognition," *Neurocomputing*, vol. 99, pp. 592–602, 2013.

[11] Jianqing Fan and Runze Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[12] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009, pp. 689–696.

[13] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang, "Fisher discrimination dictionary learning for sparse representation," in *ICCV*, 2011, pp. 543–550.

[14] Qiang Zhang and Baoxin Li, "Discriminative k-svd for dictionary learning in face recognition," in *CVPR*, 2010, pp. 2691–2698.

[15] Adrian Barbu, Yiyuan She, Liangjing Ding, and Gary Gramajo, "Feature selection with annealing for computer vision and big data learning," *IEEE Trans. PAMI*, vol. 39, no. 2, pp. 272–286, 2017.

[16] Wenjiang J Fu, "Penalized regressions: the bridge versus the lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[17] Yann LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*.