ACCURATE DICTIONARY LEARNING WITH DIRECT SPARSITY CONTROL

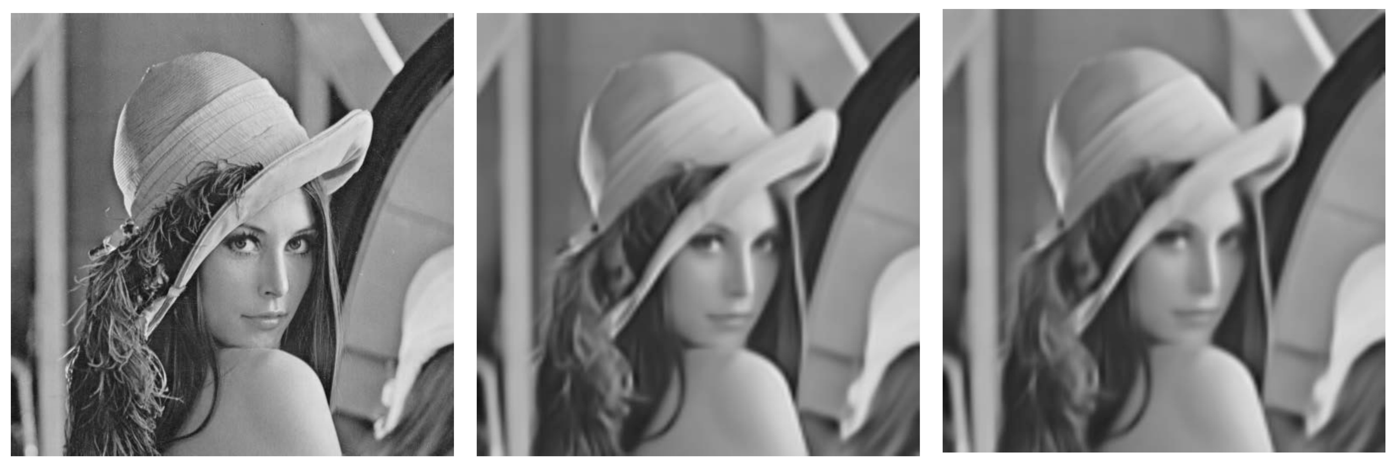
Hongyu Mou, Adrian Barbu

Florida State University, Department of Statistics

Overview

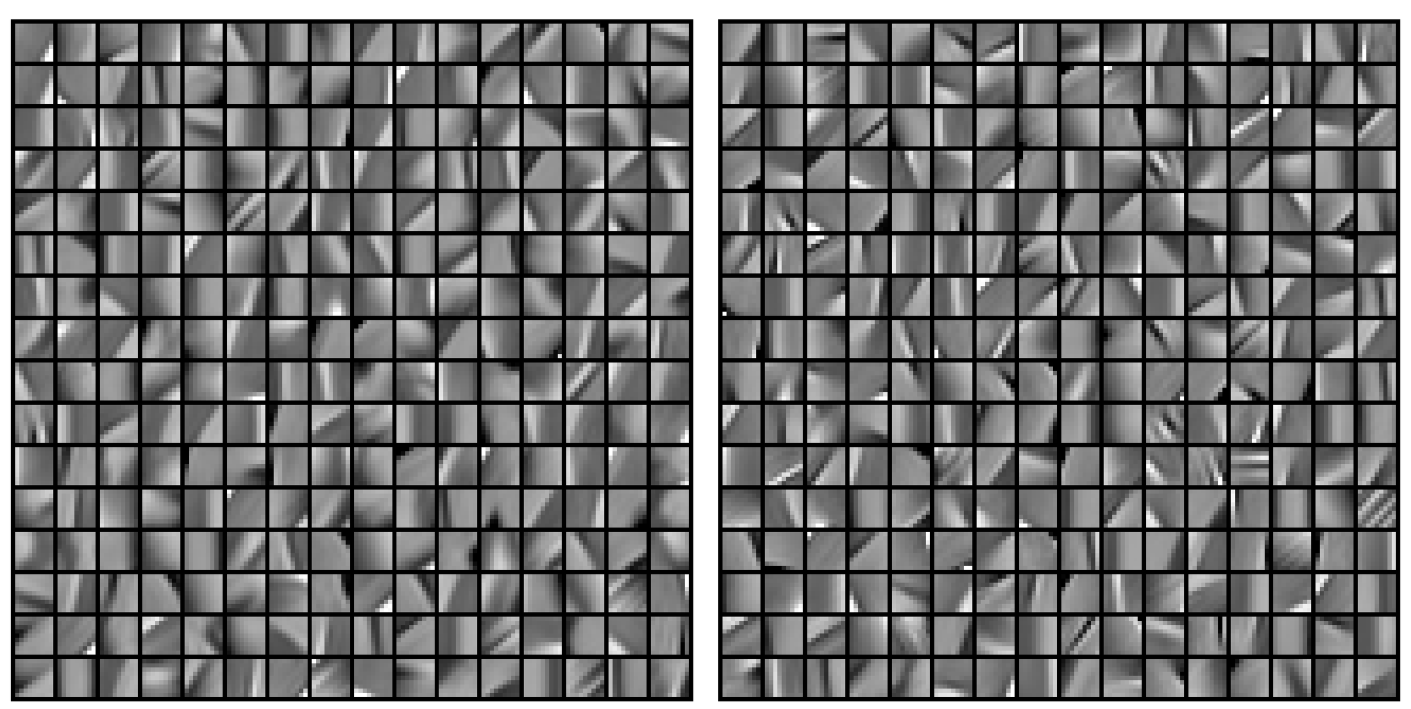
Task: find a dictionary and sparse representation for pictures

desired: direct control of the sparsity level.



Left: original picture Middle: picture reconstructed using FSA
Right: picture reconstructed using LARS

Dictionary



Left: dictionary obtained using the l_1 penalty. Right: dictionary obtained using FSA

Classical Dictionary Learning

1. Suppose we have input data $\mathbf{X}$, find a dictionary $\mathbf{D}$ and sparse coding $\mathbf{A}$:

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

2. Minimize the cost function to learn the dictionary and sparse representation:

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

3. L_1-penalized square loss function (indirect sparsity control through λ)

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

Dictionary Learning with FSA

1. Unlike using L_1-penalized loss function, we use FSA to directly control the sparsity level:

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

2. Using FSA, we have three hyper-parameters:
   η: learning rate
   μ: the speed of removing the variables
   N: number of iterations

3. We use a straightforward approach that minimizes the cost by alternately minimizing over one variable while keeping the other one fixed.

Full Algorithm

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

$$\boldsymbol{\alpha}_i = \underset{||\boldsymbol{\alpha}||_0 \le k}{\operatorname{argmin}} ||\mathbf{x}_i - \mathbf{D}_{t-1}\boldsymbol{\alpha}||_2^2$$

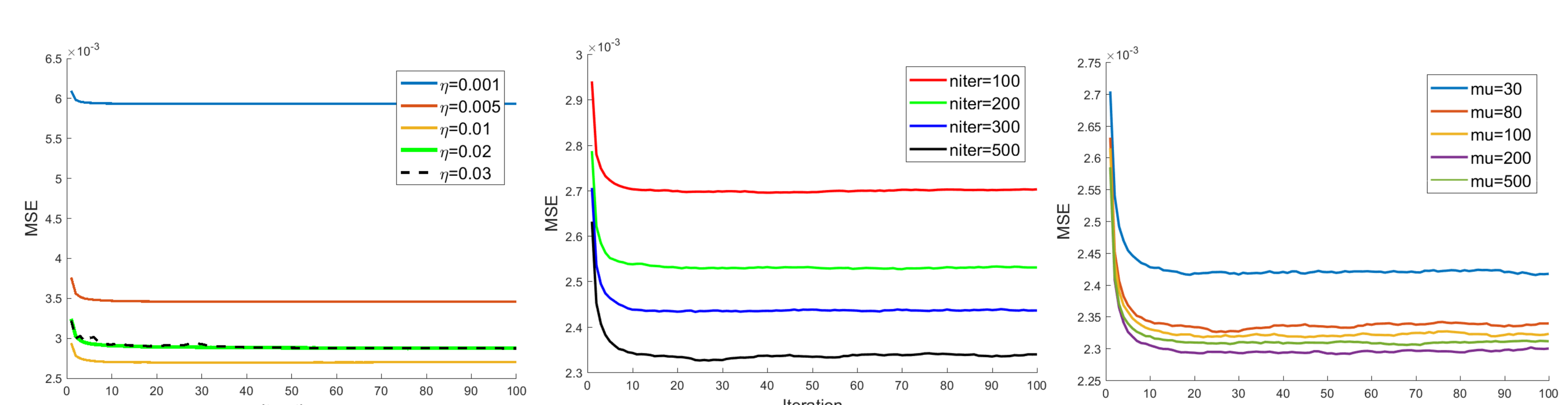
5: **end for**
6: Set $\mathbf{A} = (\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_n) \in \mathbb{R}^{p\times n}$.
7: Compute $\mathbf{D}_t$ by Algorithm 3, with $(\mathbf{X}, \mathbf{A}, \mathbf{D}_{t-1})$ as input

$$\mathbf{D}_t = \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{D}_{t-1}\boldsymbol{\alpha}_i||_2^2 \quad (5)$$

8: **end for**
9: Return $\mathbf{D} = \mathbf{D}_T$

Experiments

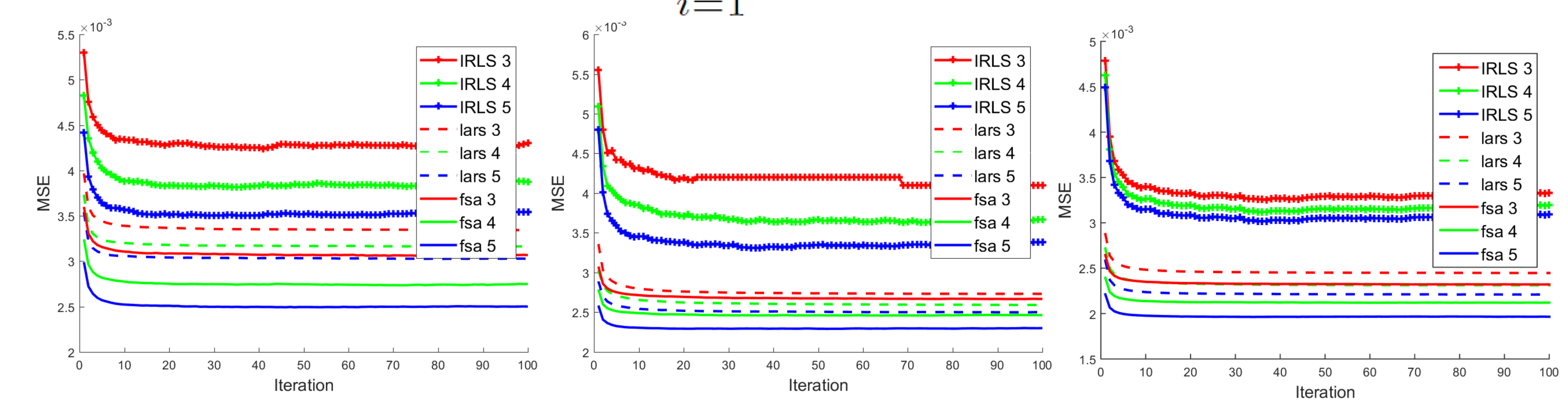
- Tune FSA hyper-parameters by comparing the mean square error.



Comparison with Other Methods

- We use the mean square error to compare the results for different sparsity methods.

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



Digit recognition application

- MNIST handwritten digit database
- Three sparse dictionary methods: LARS, IRLS, FSA (ours)
- Evaluate the classification accuracy on the sparse feature vectors using three multi-class classification methods.

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

Conclusion

- 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 $||\alpha||_0 \le k$.
- The method relies on a recent feature selection method 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.