LOCALLY LINEAR EMBEDDED SPARSE CODING FOR IMAGE REPRESENTATION

Lingdao Sha, Dan Schonfeld

University of Illinois at Chicago
Electrical and Computer Engineering
851 S Morgan St Chicago, Illinois

Jing Wang

University of Illinois at Chicago
Mathematics, Statistics and Computer Science
851 S Morgan St Chicago, Illinois

ABSTRACT

Recently, sparse coding has been widely and successfully used in image classification, noise reduction, texture synthesis and audio processing. Although traditional sparse coding method with fixed dictionaries like wavelet and curvelet can produce promising results, unsupervised sparse coding has shown its advantage by optimizing the dictionary adaptively. However, existing unsupervised sparse coding failed to consider the high dimensional manifold information within data. Recently, a graph regularized sparse coding method has shown outstanding performance by incorporating graph laplacian manifold information. In this paper, we proposed a sparse coding method called locally linear embedded sparse coding, to consider the local manifold structure as well as learning the sparse representation. We also provided a novel modified online dictionary learning method which iteratively utilizes modified least angle regression and block coordinate descent method to solve the problem. Instead of getting entire coefficient matrix then update dictionary matrix, our method updates coefficient vector and dictionary matrix in each inner iteration. Extensive experimental results have demonstrated the efficiency and accuracy of our method in image clustering.

Index Terms— Locally linear embedding, sparse coding, manifold learning, online dictionary learning, least angle regression, image clustering, SIFT

1. INTRODUCTION

Sparse coding enables successful representation of stimuli with only a few active coefficients. It has shown state-of-art results in ordinary signal processing tasks like image denoising [1] and restoration [2], audio [3] and video processing [4], as well as more complicated tasks like image classification [5] and image clustering [6]. When applied to natural images, sparse coding produces learned bases that can resemble the receptive fields of neurons in the visual cortex [7], which is similar to the results of Independent Component Analysis (ICA) [8] and Gabor filter [9]. Compared with other unsupervised methods like PCA and ICA, sparse coding can learn overcomplete basis sets and doesn't require statistical-independence of the dictionary prototype signals. In machine learning and statistics, slightly different matrix factorization problems such as non-negative matrix factorization, its variants [10] [11] and sparse principal component analysis [12] have been successfully used to obtain interpretable basis elements.

When dealing with high dimensional feature space in image clustering and classification, sparse coding with dimensionality reduction becomes a reasonable thought. Cai [13] proposed a graph regularized nonnegative matrix factorization (NMF) method, inspired by his work, Gao [14] and Zheng [6] proposed graph regularized sparse coding (GraphSC), which explicitly considers the local geometrical structure of the data. In those epic work, graph regularized NMF and sparse coding show big improvement on image clustering compared with existing NMF and sparse coding. However, all of these graph regularized work are based on graph laplacian method, which is only one of the many manifold learning methods. In this paper, we proposed a locally linear embedded sparse coding method (LLESC) together with a novel modified online dictionary learning method (MODL) to solve the objective function efficiently.

The rest of this paper is organized as follows: In Section II, we give a brief description of sparse coding problem and popular methods to solve the sparse coding problem. Section III introduces the LLESC algorithm, as well as the MODL solution. Experimental results on image clustering are presented in Section IV.

2. A BRIEF REVIEW OF SPARSE CODING

Given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_m] \in \mathbb{R}^{n\times m}$, let $\boldsymbol{D} = [\boldsymbol{d}_1, ..., \boldsymbol{d}_k] \in \mathbb{R}^{n\times k}$, where each $\boldsymbol{d}_i$ represents a basis vector in the dictionary, and $\boldsymbol{A} = [\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_m] \in \mathbb{R}^{k\times m}$ be the coefficient matrix, where each column is a sparse representation for a data point. A good dictionary and coefficient pair should minimize the empirical loss function, which can be represented as $\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_p$. The typical norms used for measuring the loss function are the L_p norms where $p = 1, 2$ and ∞. Here we concentrate on least square loss problems when $p = 2$.

The objective function of sparse coding can be formulated as:

$$\min_{\boldsymbol{D},\boldsymbol{A}} \|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{A}\|_F^2 + \beta \sum_{i=1}^{m} f(\boldsymbol{\alpha}_i), \quad \text{s.t. } \|\boldsymbol{d}_i\|^2 \leqslant c, i=1,...,k \tag{1}$$

where f is a function to measure the sparseness of $\boldsymbol{\alpha}_i$ and $\|\cdot\|_F$ denotes the matrix Frobenius norm.

Following [15] [16], we adopt the idea of L_1 norm instead of L_0, which can produce similar results with affordable computational cost. The objective function then becomes:

$$\min_{\boldsymbol{D},\boldsymbol{A}} \|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{A}\|_F^2 + \beta \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1, \quad \text{s.t. } \|\boldsymbol{d}_i\|^2 \leqslant c, i=1,...,k \tag{2}$$

Although the objective function is not convex with $\boldsymbol{D}$ and $\boldsymbol{A}$ together, it is convex with either one fixed. We iteratively optimize the objective function by minimizing over one variable with the other one fixed. Thus, it becomes an L_1-regularized least squares problem with an L_2-constrained least square problem.

3. LOCALLY LINEAR EMBEDDED SPARSE CODING (LLESC)

3.1. Algorithm Outline

Locally linear embedding (LLE) is an unsupervised learning algorithm that computes low dimensional, neighborhood preserving embedding of high dimensional data. LLE attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstruction [17]. Given a set of m dimensional data points $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_m$, we can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are:

$$\tfrac{1}{2}\sum_{i=1}^{m} |\boldsymbol{x}_i - \textstyle\sum_j W_{ij}\boldsymbol{x}_j|^2 = Tr(\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T) \tag{3}$$

Where $\boldsymbol{L} = (\boldsymbol{I}-\boldsymbol{W})^{\boldsymbol{T}}(\boldsymbol{I}-\boldsymbol{W})$, $\boldsymbol{I}$ is identity matrix, $\boldsymbol{W}$ is weight matrix.

Nearest neighbor is a necessary step to compute the weight matrix. Besides using Euclidean distance, we also utilizes scale-invariant feature transform (SIFT) [18] for nearest neighbor calculation, which shows better performance in situations with scaled and rotated image objects.

LLE constructs a neighborhood preserving mapping: $\boldsymbol{x}_i \mapsto \boldsymbol{\alpha}_i$. By incorporating the LLE regularizer into the original sparse coding, we can get the following objective function of LLESC:

$$\min_{\boldsymbol{D},\boldsymbol{A}} \|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{A}\|_F^2 + \lambda Tr(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^T) + \beta\sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \quad \text{s.t. } \|\boldsymbol{d}_i\|^2 \leqslant c, i=1,...,k \tag{4}$$

where $\lambda \geqslant 0$ is the regularization parameter.

3.2. Coefficients Learning and Dictionary Learning

In this section, we show how to solve problem (4) with modified online dictionary learning algorithm.

Fixing dictionary $\boldsymbol{D}$, the objective function becomes:

$$\min_{\boldsymbol{A}} \|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{A}\|_F^2 + \lambda Tr(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^T) + \beta\sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \tag{5}$$

As problem (5) is convex, global minimum can be achieved[19].

With modified online dictionary learning, we update each vector $\boldsymbol{\alpha}_i$ individually, while keeping all the other vectors constant. In order to solve the problem by optimizing over each $\boldsymbol{\alpha}_i$, we rewrite problem (5) in vector form.

Reconstruction error $\|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{A}\|_F^2$ can be written as:

$$\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|^2 \tag{6}$$

As matrix $\boldsymbol{L}$ is symmetric in LLE, the regularizer $Tr(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^T)$ can be rewritten as:

$$Tr(\boldsymbol{A}\boldsymbol{L}\boldsymbol{A}^T) = Tr(\sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j^T) = \sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j \tag{7}$$

We combine reconstruction error with LLE regularizer, add sparsity constrain to it, the objective function becomes:

$$\min_{\boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x_i} - \boldsymbol{D}\boldsymbol{\alpha}_i\|^2 + \lambda \sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j + \beta\sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \tag{8}$$

When updating $\boldsymbol{\alpha}_i$, the other vectors $\{\boldsymbol{\alpha}_j\}_{j\neq i}$ are fixed[6] [20]. Thus, we get the following optimization problem:

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|^2 + \lambda L_{ii}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i^T\boldsymbol{h}_i + \beta\sum_{j=1}^{k} |\boldsymbol{\alpha}_i^{(j)}| \tag{9}$$

Where $\boldsymbol{h}_i = 2\lambda(\sum_{j\neq i} L_{ij}\boldsymbol{\alpha}_j)$ and $\boldsymbol{\alpha}_i^{(j)}$ is the j-th coefficient of α_i

In Algorithm 1 of modified online dictionary learning (MODL), we keep dictionary $\boldsymbol{D}$ fixed, optimizing each individual coefficient α_i with all other coefficients fixed for each input data $\boldsymbol{x_i}$. The method used is modified least angle regression which will be explained in algorithm 2. Dictionary update is by block coordinate descent method, please reference [20] for detail.

Algorithm 1: MODL

Require: $\boldsymbol{x} \in \mathbb{R}^m$ from $\boldsymbol{p}(\boldsymbol{x})$ ($\boldsymbol{x}$ sequentially aligned in $\boldsymbol{p}(\boldsymbol{x})$), $\beta \in \mathbb{R}$ (regularization parameter), $\boldsymbol{D}_0 \in \mathbb{R}^{m\times k}$ (initial dictionary), T (number of samples in data set $\boldsymbol{p}(\boldsymbol{x})$).

1: $\boldsymbol{A}_0 \in \mathbb{R}^{k\times k} \leftarrow 0$, $\boldsymbol{B}_0 \in \mathbb{R}^{m\times k} \leftarrow 0$ (Reset the "past"information)

2: **for** $t = 1$ **to** T **do**

3: Draw $\boldsymbol{x_t}$ from $\boldsymbol{p}(\boldsymbol{x})$ (sequentially drawn)

4: Sparse coding: compute using modified LARS (Algorithm 2)

$$\boldsymbol{\alpha}_t \triangleq \arg\min_{\boldsymbol{\alpha}\in\mathcal{R}^k} \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda L_{tt}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{h}_t + \beta\|\boldsymbol{\alpha}\|_1$$

5: $\boldsymbol{A}_t \leftarrow \boldsymbol{A}_{t-1} + \boldsymbol{\alpha}_t\boldsymbol{\alpha}_t^T$

6: $\boldsymbol{B}_t \leftarrow \boldsymbol{B}_{t-1} + \boldsymbol{x}_t\boldsymbol{\alpha}_t^T$

7: Compute $\boldsymbol{D}_t$ using block coordinate descent method [20], with $\boldsymbol{D}_{t-1}$ as warm restart, so that

$$\boldsymbol{D}_t \triangleq \arg\min_{\boldsymbol{D}\in\mathcal{C}} \frac{1}{t}\sum_{i=1}^{t} (\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i,j=1}^{m} L_{ij}\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j$$

$$+\beta\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_1)$$

$$= \arg\min_{\boldsymbol{D}\in\mathcal{C}} \frac{1}{t}(\frac{1}{2}Tr(\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{A}_t) - Tr(\boldsymbol{D}^T\boldsymbol{B}_t))$$

8:**end for**

9:Return $\boldsymbol{D}_T$, $\boldsymbol{A}$ for complete dictionary and coefficients learning

---

Notation: $\mathcal{C} \triangleq \{\boldsymbol{D} \in \mathbb{R}^{m\times k} s.t. \forall j = 1, ..., k, \boldsymbol{d}_j^T\boldsymbol{d}_j \leqslant 1\}$.

3.3. Modified Least Angle Regression

Least Angle Regression (LARS) [21] is a regression method that provides a general version of forward selection, which is highly efficient in solving LASSO [22]. We follow the steps presented in [23]. In step 7 of Algorithm 2, instead of calculating the ordinary least square solution (OLS) (10), we calculate the locally linear embedded least square solution (LLELS) (11) to incorporate structure information.

$$\boldsymbol{\alpha}_{OLS}^{(k+1)} = (\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{D}_{\mathcal{A}})^{-1}\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{y} \quad (10)$$

$$\boldsymbol{\alpha}_{LLELS}^{(k+1)} = (\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{D}_{\mathcal{A}} + \lambda L_{kk}\boldsymbol{I})^{-1}(\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{x} - \boldsymbol{h}_k/2) \quad (11)$$

Where $\boldsymbol{I}$ is identity matrix and $\boldsymbol{h}_k = 2\lambda(\sum_{k\neq j} L_{kj}\boldsymbol{\alpha}_j)$.

Algorithm 2: Modified Least Angle Regression

1: Initialize the coefficient vector $\boldsymbol{\alpha}^{(0)} = 0$ and the fitted vector $\hat{\boldsymbol{x}}^{(0)} = 0$.

2: Initialize the active set $\mathcal{A} = \phi$ and the inactive set $\mathcal{I} = 1, ..., p$.

3: **for** $k = 0$ **to** $p - 2$ **do**

4: Update the residual $\boldsymbol{\varepsilon} = \boldsymbol{x} - \hat{\boldsymbol{x}}^{(k)}$

5: Find the maximal correlation $c = max_{i\in\mathcal{I}}|\boldsymbol{d}_i^T\boldsymbol{\varepsilon}|$

6: Move variable corresponding to c from $\mathcal{I}$ to $\mathcal{A}$

7: Calculate the graph constrained least square solution:
$\boldsymbol{\alpha}_{LLELS}^{(k+1)} = (\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{D}_{\mathcal{A}} + \lambda L_{kk}I)^{-1}(\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{x} - h_k/2)$
Where I is identity matrix and $h_k = 2\lambda(\sum_{k\neq j} L_{kj}\boldsymbol{\alpha}_j)$

8: Calculate the current direction: $\mathtt{d} = \boldsymbol{D}_{\mathcal{A}}\boldsymbol{\alpha}_{LLELS}^{(k+1)} - \hat{\boldsymbol{x}}^{(k)}$

9: Calculate the step length:
$\gamma = \min_{i\in\mathcal{I}}^{+}\{\frac{\boldsymbol{d}_i^T\varepsilon - c}{\boldsymbol{d}_i^T\mathtt{d} - c}, \frac{\boldsymbol{d}_i^T\varepsilon + c}{\boldsymbol{d}_i^T\mathtt{d} + c}\}, 0 \leq \gamma \leq 1$

10: Update regression coefficients:
$\boldsymbol{\alpha}^{(k+1)} = (1-\gamma)\boldsymbol{\alpha}^{(k)} + \gamma\boldsymbol{\alpha}_{LLELS}^{(k+1)}$

11: Update the fitted vector $\hat{\boldsymbol{x}}^{(k+1)} = \hat{\boldsymbol{x}}^{(k)} + \gamma\mathtt{d}$

12:**end for**

13:Let $\boldsymbol{\alpha}^{(p)}$ be the full graph constrained least square solution
$\boldsymbol{\alpha}^{(p)} = (\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{D}_{\mathcal{A}} + \lambda L_{(p-1)(p-1)}\boldsymbol{I})^{-1}(\boldsymbol{D}_{\mathcal{A}}^T\boldsymbol{x} - h_{p-1}/2)$
where $\boldsymbol{I}$ is identity matrix and $h_{p-1} = 2\lambda(\sum_{p-1\neq j} L_{(p-1)j}\boldsymbol{\alpha}_j)$

14: Output: the series of coefficients $\boldsymbol{A} = [\boldsymbol{\alpha}^{(0)}, ..., \boldsymbol{\alpha}^{(p)}]$

Note: $\boldsymbol{d}_i$ is column of Dictionary $\boldsymbol{D}$, $\mathtt{d}$ is direction.

4. EXPERIMENTAL RESULTS

In this section, we present image clustering experiments on CMU-PIE and COIL data set [1], data statistics are shown in table 1. We compared clustering accuracy of our method (LLESC) against several unsupervised methods. We also compared the computation efficiency between LLESC and GrapSC methods [14] [6]. All clustering tasks are based on a Windows 10 machine with Intel Core i7-2820M 2.3GHz CPU and 16GB RAM. Algorithms were implemented and executed in MATLAB environment. We used VLFeat toolbox [2]for SIFT calculation.

We use both PCA and K-SVD for preprocessing (pick the best results), after getting the coefficient matrix ($\boldsymbol{A}$) by GraphSC and LLESC, K-means method will be used to cluster those coefficients. We use computation time from matlab as efficiency evaluation metric, normalized mutual information (NMI) [13] [6] as clustering accuracy evaluation metric.

Table 2, figure 1, table 3 and figure 2 shows LLESC clustering results on CMU-PIE and COIL data set. Figure 3 shows an example of SIFT matching of two images with different orientations. Figure 4 and figure 5 show LLESC and LLESCsift (LLESC with SIFT) clustering results with different regularization parameter λ and number of clusters k on CMU-PIE and COIL data set. We can easily find LLESCsift performances slightly better than LLESC on COIL, as COIL data set contains images with different orientations and SIFT is better than Euclidean in finding similar images in those data sets. Finally, figuer 6 shows our LLESC with MODL algorithm is more efficient than GraphSC in clustering on CMU-PIE and COIL data set.

Table 1: Statistics of the data set

Data set	Size(N)	Dimensionality (M)	# of class (K)
CMU-PIE	1428	1024	68
COIL20	1440	1024	20

Table 2: Clusetering performance on CMU-PIE (K is number of clusters)

K	Normalized Mutual Information (%)				
	Kmeans	**PCA**	**KSVD**	**SC**	**LLESC**
4	33 ± 5.6	44 ± 6.2	100	100	100
12	52 ± 4.8	55 ± 5.1	91 ± 2.2	95 ± 1.2	97 ± 1.1
20	55 ± 3.3	59 ± 4.5	75 ± 2.6	91 ± 1.1	96 ± 1.3
28	59 ± 3.7	60 ± 3.4	76 ± 2.8	90 ± 1.2	96 ± 1.1
36	60 ± 3.9	63 ± 1.6	77 ± 3.1	88 ± 2.3	95 ± 1.2
44	60 ± 2.4	65 ± 1.1	74 ± 2.7	85 ± 1.5	95 ± 1.1
52	61 ± 2.2	62 ± 1.9	76 ± 2.2	83 ± 2.1	94 ± 1.3
60	65 ± 3.5	66 ± 2.1	78 ± 1.9	80 ± 1.4	94 ± 1.0
68	63	66	75	77	93

[1]http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

[2]http://www.vlfeat.org/

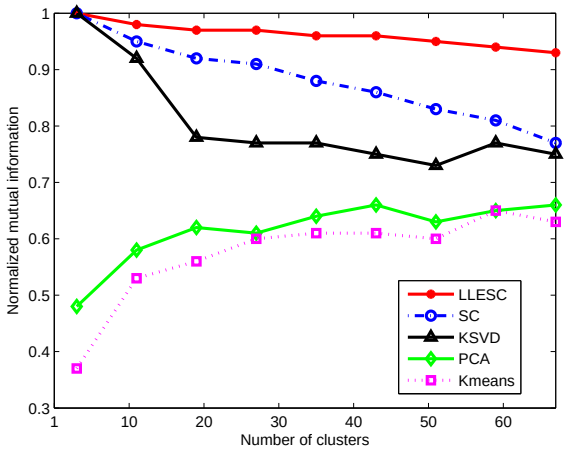


Fig. 1: Normalized mutual information versus the number of clusters on CMU-PIE data set

Table 3: Clusetering performance on COIL20 (K is number of clusters)

	Normalized Mutual Information (%)				
K	**Kmeans**	**PCA**	**KSVD**	**SC**	**LLESC**
2	67 ± 8.5	54 ± 9.1	66 ± 8.8	81 ± 5.2	83 ± 9.6
4	65 ± 8.3	63 ± 9.2	64 ± 6.9	84 ± 6.3	84 ± 9.9
6	66 ± 9.4	59 ± 8.1	70 ± 8.1	78 ± 4.3	83 ± 9.2
8	61 ± 8.6	61 ± 9.7	79 ± 6.4	82 ± 5.2	79 ± 8.9
10	59 ± 9.6	60 ± 7.9	72 ± 5.5	84 ± 2.1	80 ± 8.6
12	62 ± 7.9	69 ± 6.5	70 ± 4.6	82 ± 2.4	81 ± 8.4
14	66 ± 7.7	65 ± 6.7	69 ± 5.1	76 ± 2.9	83 ± 6.3
16	71 ± 6.5	61 ± 5.5	72 ± 2.3	81 ± 3.3	82 ± 6.7
18	70 ± 4.4	60 ± 4.9	71 ± 1.4	76 ± 1.6	78 ± 5.9
20	72.4	66.7	74.1	77.3	80.3

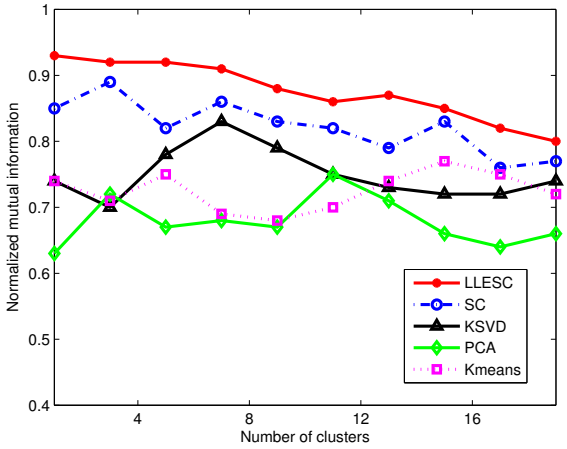


Fig. 2: Normalized mutual information versus the number of clusters on COIL20 data set

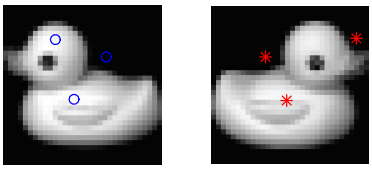
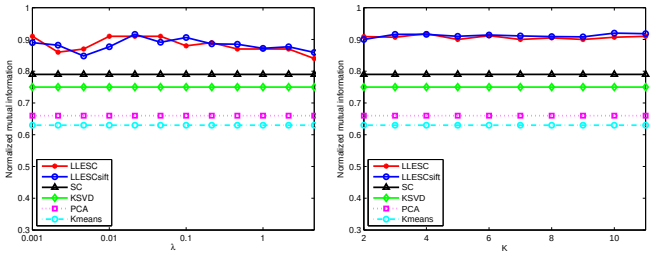
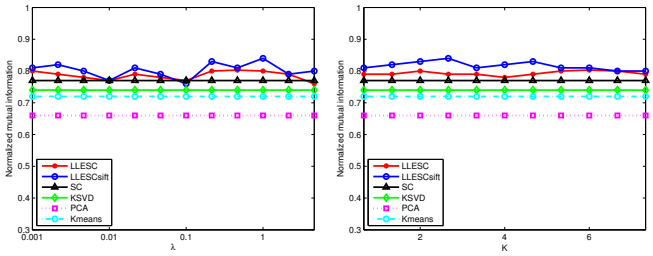


Fig. 3: SIFT matching example



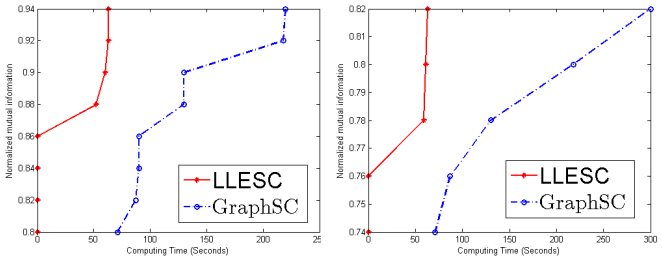
(a) NMI vs λ (b) NMI vs k

Fig. 4: Clustering performance with different values of regularization parameter (λ) and the number of nearest neighbors (k) on CMU-PIE face database.



(a) NMI vs λ (b) NMI vs k

Fig. 5: Clustering performance with different values of regularization parameter (λ) and the number of nearest neighbors (k) on COIL20 face database.



(a) On CMU-PIE data set (b) On COIL20 data set

Fig. 6: Clustering time between LLESC and GraphSC on CMU-PIE and COIL20 data set.

5. REFERENCES

[1] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," vol. 15, no. 12, pp. 3736–3745, 2006.

[2] Julien Mairal, Julien Mairal, Michael Elad, Michael Elad, Guillermo Sapiro, and Guillermo Sapiro, "Sparse representation for color image restoration," in *the IEEE Trans. on Image Processing*. 2007, pp. 53–69, ITIP.

[3] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng, "Grosse et al. 149 shift-invariant sparse coding for audio classification," .

[4] Bruno A. Olshausen, "Sparse coding of time-varying natural images," in *IN PROC. OF THE INT. CONF. ON INDEPENDENT COMPONENT ANALYSIS AND BLIND SOURCE SEPARATION*, 2000, pp. 603–608.

[5] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *in IEEE Conference on Computer Vision and Pattern Recognition(CVPR*, 2009.

[6] Miao Zheng, Jiajun Bu, Chun Chen, Can Wang, Lijun Zhang, Guang Qiu, and Deng Cai, "Graph regularized sparse coding for image representation," *IEEE Transactions on Image Processing*, pp. 1327–1336, 2011.

[7] D. J. Fieldt B. A. Olshausen, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[8] Anthony J. Bell and Terrence J. Sejnowski, "The "independent components" of natural scenes are edge filters," 1997.

[9] S. Marelja, "Mathematical description of the responses of simple cortical cells," *Journal of the Optical Society of America*, vol. 70, pp. 1297–1300, 1980.

[10] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*. 2001, pp. 556–562, MIT Press.

[11] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Jour. of*, pp. 1457–1469, 2004.

[12] Hui Zou, Trevor Hastie, and Robert Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, pp. 2006, 2004.

[13] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 33, no. 8, pp. 1548–1560, 2011.

[14] Shenghua Gao, Ivor Wai hung Tsang, Liang tien Chia, and Peilin Zhao, "Local features are not lonely laplacian sparse coding for image classification," .

[15] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[16] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," 1995.

[17] Lawrence K. Saul and Sam T. Roweis, "An introduction to locally linear embedding," Tech. Rep., 2000.

[18] David G. Lowe, "Distinctive image features from scale-invariant keypoints," 2003.

[19] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.

[20] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," 2010.

[21] Iain Johnstone Robert Tibshirani Bradley Efron, Trevor Hastie, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–451, june 2004.

[22] Saharon Rosset and Ji Zhu, "Piecewise linear regularized solution paths," *Ann. Statist*, p. 1030, 2007.

[23] Rasmus Larsen Bjarne Ersboll Karl Sjostrand, Line H. Clemmensen, "Spasm: A matlab toolbox for sparse statistical modeling," *Journal of Statistical Software*, 2010.