Dictionary learning Based on Laplacian Score in Sparse Coding

Jin Xu and Hong Man

Department of Electrical and Computer Engineering, Stevens institute of Technology, Hoboken, NJ 07030 USA

Abstract. Sparse coding, which is represented a vector based on sparse linear combination of a dictionary, is widely applied on signal processing, data mining and neuroscience. How to get a proper dictionary is a problem, which is data dependent and computational cost. In this paper, we treat dictionary learning in the unsupervised learning view and proposed Laplacian score dictionary (LSD). This new method uses local geometry information to select atoms for dictionary. Comparison experiments with competitive clustering based dictionary learning methods are established. We also compare LSD with full-training-data-dictionary and others classic methods in the experiments. The results on binary classes datasets and multi class datasets from UCI repository demonstrate the effectiveness and efficiency of our method.

Keywords: Sparse coding, Unsupervised learning, Clustering, Dictionary learning, Laplacian Score

1 Introduction

Sparse coding (sparse representation) has been improved extensively recently. In sparse coding, a signal y is represented by combination of a defined dictionary D atoms, which aims to use the least number of atoms. Sparse coding has a lot of applications. Such as classification, image denosing and online learning.

In sparse representation, computational complexity is a problem. There are three methods to reduce the computation of sparse coding. (1) Proper dictionary: The dictionary represents the key information for the data, and size of dictionary can affect the computation significantly. How to get the ideal dictionary is the focus of this paper. (2) Dimension reduction: This method can remove the redundant features of data for sparse coding and it was successful in some applications. (3) Algorithm: This method is to use different optimization methods to speed up the sparse representation.

The successful sparse coding lies in the proper dictionary. How to get the proper dictionary is a hot research topic recently. At first, the pre-constructed dictionaries are chosen, such as steerable wavelets, curvetted, DCT matrix, and more. Then a tunable selection for dictionary is applied, such as wavelet packets and bandelets. Recently, more research are focus on the learning the dictionary

from the examples. Unsupervised learning methods are also shown success in dictionary acquiring for sparse coding.

Laplacian score (LS) was proposed for unsupervised feature selection. In their work, LS is to evaluate the locality preserving ability and to rank the feature power for the feature selection. LS has successful application in the face recognition and semi-supervised learning. In this paper, the work is focus on the dictionary learning via unsupervised learning. The key idea is to use LS to evaluate the locality preserving ability of training data, and the higher ranked data is selected as atoms for the dictionary in sparse coding. This method is compared with classic clustering methods such as self-organized map (SOM) and neural gas (NGAS) in dictionary learning to show effectiveness.

The contribution of this work is listed:

- An unsupervised learning methods based on LS for dictionary learning in sparse representation is presented. While most of LS applications has been concentrated on features view, and we first proposed atoms selection based LS criteria.
- The proposed dictionaries are applied to both binary and mutli-class UCI databsets, the results are shown with classification results, reconstruction results.
- Competitive experiments results are obtained on diverse UCI datasets, providing the capabilities of proposed dictionary learning methods.

The rest of paper is organized as follows: Section 2 presents related work and sparse representation based classification. Section 3 presents Laplacian score (LS) criteria for dictionary learning. Section 4 presents the comparison experimental results with the UCI datasets. Finally, section 5 concludes the paper and discusses some future planning.

2 Related Work

Recently, sparse representation based classification (SRC) was proposed and shown successful application in face classification. SRC utilizes category based reconstruction error to classify testing data. The performance of SRC can be used to evaluate the dictionary in sparse coding.

In sparse coding, a dictionary containing a set of n atoms (data vectors) is defined as $\mathbf{D} = [a_1^1, \cdots, a_1^{n_1}, \cdots, a_c^1, \cdots, a_c^{n_c}]$, where $\mathbf{D} \in \mathbb{R}^{m \times n}$, c is class label for each atom, n_i is the number of atoms associated with class i. The intention of sparse representation is to code a new test vector y with the form

$$y = Dx \in \mathbb{R}^m \tag{1}$$

where $x = [0, \cdots, 0, \alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,n_i}, 0, \cdots, 0]^T \in \mathbb{R}^n$. x should be sparse with the least number of nonzero. Normally, ℓ_1-regularized least squares method [**?**] is used to solve this problem.

$$\widehat{x} = arg\min\{\|y - Dx\|_2^2 + \lambda\|x\|_1\} \tag{2}$$

In SRC, it utilizes the representation residual to judge the target class [**?**]. For each class i, a function is defined as $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$, which chooses the coefficients associated to i-th class. Then classification process can be shown as:

$$label(y) = arg\min r_i(y), \quad r_i(y) = \|y - D\delta_i(x)\|^2 \tag{3}$$

Algorithm 1 Sparse Representation based on Classification

1: **Input:** a dictionary $D \in \mathbb{R}^{m\times n}$ with c classes, a test data $y \in \mathbb{R}^m$
2: Solve ℓ_1-regularized least squares problem:
$\widehat{x} = arg\min\{\|y - Dx\|_2^2 + \lambda\|x\|_1\}$
3: Compute the residuals:
$r_i(y) = \|y - D\delta_i(x)\|^2 \quad for \quad i = 1, \cdots, c$
4: **Output:** $label(y) = arg\min r_i(y)$

The SRC process is shown in Algorithm 1. A case study is for SRC is shown in Fig.1. "Libreas Movement" datasets from UCI Repository are used. A dictionary is trained with Laplacian score criteria, which has 65 atoms from 13 classes. Then the dictionary is applied on a test data to get the sparse vector for atoms. The left figure shows the details of sparse coding coefficients. The construction residuals based on each class are shown in the right figure. It obvious to see the residual for class 6 is smallest. Then the SRC algorithm will put the test data to the class 6. Actually, the total classes for the "Libreas Movement" datasets are 15. However, there are just 13 classes data are chosen in this case study. How to leverage all category data in the dictionary is a problem in dictionary construction.

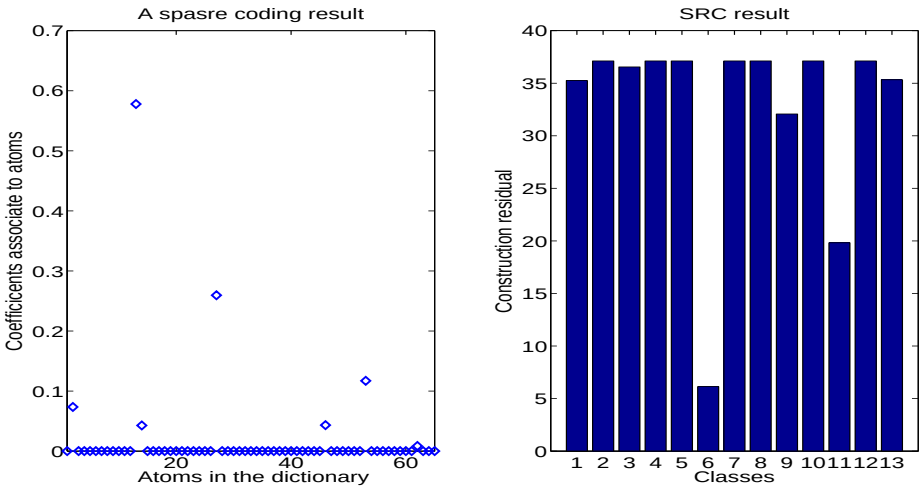

Fig. 1. Classification result for data Glass

In dictionary learning, a popular criteria is optimized with sparse coding. The dictionary updating and sparse coding are processed iteratively until some threshold are reached. Though it has been proved effectiveness in some application, the iterative process has made heavy computation cost. The following methods are from this view.

MOD is frame design algorithm called the method of optimal directions (MOD) [**?**]. In the MOD, first, a dictionary is initialed, then sparse coefficients of a signal are calculated from the dictionary. Then use the coefficients and the original signal to update the dictionary. The stop criteria is based on the least square error.

KSVD [**?**] is different with mod that the atoms in the dictionary are updated sequentially. It is relative to the k-means as it tries to update the atoms based on associate examples. In KVSD, first, find the group examples for the atoms. Then calculate residuals for the chosen examples. Finally, use singular value decomposition based on the residuals to update the dictionary atoms.

Efficient sparse coding algorithms [**?**]is based on the iteratively solving two least square optimization issue: ℓ_1 norm regularized and ℓ_2 norm constrained. The problem can be written as:

$$\min_{\alpha,D} \frac{1}{2\sigma^2}\|D\alpha - X\|^2 + \lambda\|\alpha\|_1 \quad subject\ \ to\ \ \Sigma D \leq c \tag{4}$$

In the learning process, this method optimizes dictionary D or sparse coefficients α when fixed other. This method is kind of acceleration of the sparse coding process which can be applied in the large databases.

Supervised dictionary learning[**?**] tried to combine the different categories information to the sparse coding process. The formulation shows how it works.

$$\min_{\alpha,\theta,D} (\Sigma_i C(y_i f(x_i, \alpha_i, \theta)) + \lambda_0\|D\alpha - X\|^2 + \lambda_1\|\alpha\|_1) + \lambda_2\|\theta\|_2^2 \tag{5}$$

where C is kind of loss function which is similar to the loss function of SVM. $\theta, \lambda_0, \lambda_1, \lambda_2$ are regularization paramether. The loss function utilizes label information in the optimization process.

3 Laplacian Score for Dictionary Learning

Laplacian score evaluates local geometrical structures without data labels information. This method is based on Laplacian Eigenmaps and Locality Preserving Projection.

Given a data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$, where $\mathbf{X} \in \mathbb{R}^{m\times n}$. The feature vectors for the data set are $F = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_m\}$. Assume S_r is the Laplacian score for the rth sample $\mathbf{x}_r$, $r = 1, \cdots, n$. The Laplacian score based on each sample can be stated as follows:

1. A nearest neighbor graph G is constructed with different feature vectors ($\mathbf{f}_i$ and $\mathbf{f}_j$, $i = 1, \cdots, m$). In details, if feature vector $\mathbf{f}_i$ is among k nearest neighbors of $\mathbf{f}_j$, $\mathbf{f}_i$ and $\mathbf{f}_j$ are defined connected in graph G.

2. The weight matrix of graph G is S_{ij}. When $\mathbf{f}_i$ and $\mathbf{f}_j$ are connected, $S_{ij} = e^{-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{t}}$, otherwise $S_{ij} = 0$. Where t is a suitable constant.

3. Then S_r for each sample can be caculated as

$$S_r = \frac{\tilde{\mathbf{x}}_r^T L \tilde{\mathbf{x}}_r}{\tilde{\mathbf{x}}_r^T D \tilde{\mathbf{x}}_r} \tag{6}$$

where $D = diag(S\mathbf{1}), \mathbf{1} = [1, \cdots, 1]^T, L = D - S$, and $\tilde{\mathbf{x}}_r$ is calculated via:

$$\tilde{\mathbf{x}}_r = \mathbf{x}_r - \frac{\mathbf{x}_r^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}} \tag{7}$$

The results of S_r are used to choose atoms for the dictionary, which aims to effectively utilize graph structure information.

SOM and NGAS are classic cluster methods in the unsupervised learning. SOM and NGAS can preserve the topological properties of training data, which is the inspiration of our work. The center trained by SOM and NGAS are used as atoms in dictionary for sparse coding. Relative work has shown some application between NGAS and sparse coding. It is important to mentioned that class label are needed for atoms. In this work, the atoms trained by SOM and NGAS are labeled based on 5-nearest neighbor training data voting.

4 Experiment

In this section, we present experiments on different UCI datasets, especially on multi-category datasets which are more challenge in application. The experiment is aimed to show the effectiveness of proposed dictionary method. The results are presented with SRC classification accuracy and data reconstruction errors.

4.1 Experiment Datasets

Six UCI datasets are chosen in the experiments. The details are shown in Table 1. Data "Car Evaluation" and "Tic-Tac-Toe" are binary datasets. Data "Contraceptive Method Choice" has three classes. The rest are typical multi-class datasets which are frequently used in the data mining. The feature number and the data size are also shown in Table 1.

4.2 Experiment Setup

In the experiments, 5-fold cross-validation are applied on each datasets for comparison of different learning models. The training data are trained with LS, SOM and NGAS to get dictionaries. We set the different sizes of dictionary to shown the performances. In detail, the sizes of dictionary is rang from 10% to 50% of training data size. Then SRC classifier, according to Algorithm 1, is applied on testing data to evaluate the performance of different dictionaries. In order to

Table 1. UCI Experiment Data Sets

Name	Feature number	Total size	Class number
Car Evaluation	6	1728	2
Tic-Tac-Toe	9	958	2
Contraceptive Method Choice	9	1473	3
Glass	10	214	7
Image Segmentation	19	2310	7
Libras Movement	90	360	15

get comprehensive results, three references are introduced: LibSVM, k-nearest neighbors classifier and full-training-data-dictionary. It is important to point out that all three references are based on entire training data. It is not equivalent for proposed dictionary learning method to reach same performance with these three references.

For simplicity, the dictionaries trained with LS, SOM and NGAS are abbreviated as LSD, SOMD and NGASD. The LibSVM classifier is simplified as SVM. K-nearest neighbors classifier are simplified as KNN5, which is evaluated by 5 neighbors in our experiments. The full-training-data-dictionary is abbreviated as ALLIND.

In sparse coding, reconstruction error for each test data is represented as

$$error_y = \sqrt[2]{\|y - Dx\|^2} \tag{8}$$

and can be evaluated the quality of dictionary. In our experiments, the average reconstruction error among LSD, SOMD, NGASD and ALLIND are shown in different dictionary sizes.

The sparse coding tool is from Stanford. The libSVM classifer is from Support Vector Machines Library. When the datasets are multi-categories, LibSVM use "1-v-r" method to transfer mulit-class issue to binary issue. SOM and NGAS training methods are from SOM TOOLBOX.

4.3 Experiment Result

We perform our comparison models on different datasets and results are shown in the following.

Fig. 2 shows the results on the data "Car Evaluation". From the classification view, LSD has higher classification accuracy compared with SOMD and NGASD. The accuracy of LSD is increased with dictionary size enlarging. LSD performance reaches the equal level with SVM when dictionary size is 20% of training data, and LSD has almost same accuracy with ALLIND when dictionary size reach 50% of training data. In the right figure, the reconstruction error from LSD decreases with more atoms in dictionary.

The performance of data "Tic-Tac-Toe " is shown in Fig. 3. In the left figure, LSD has better performance than SVM and it can surpass ALLIND performance when dictionary size is larger than 40% of training data. The SOMD performance

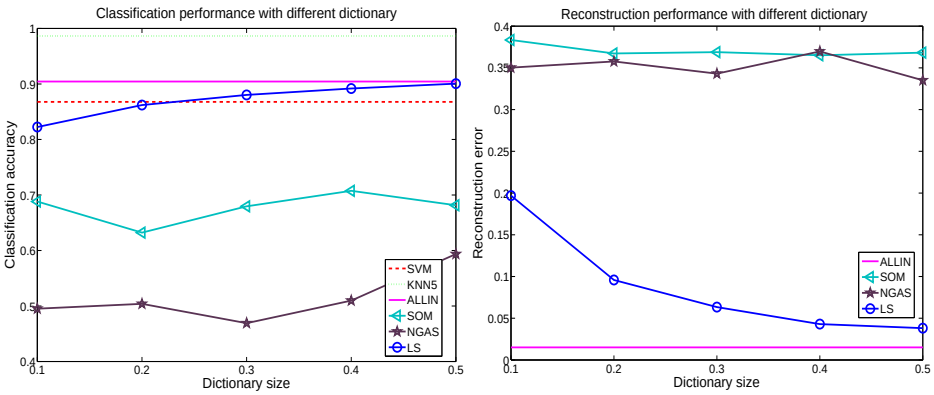

Fig. 2. Left : Classification comparison with data "Car Evaluation " **Right** : Reconstruction error with data "Car Evaluation "

can be competitive with SVM. In the reconstruction performance, NGASD has better rate than ALLIND which is interesting point.

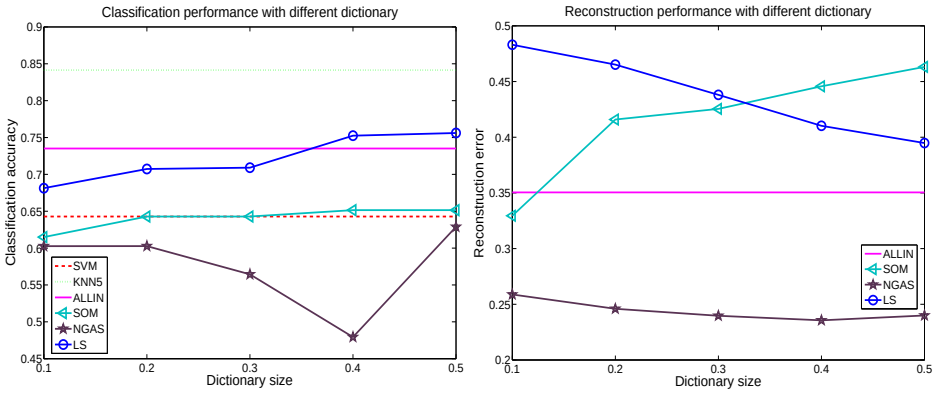

Fig. 3. Left : Classification comparison with data "Tic-Tac-Toe " **Right** : Reconstruction error with data "Tic-Tac-Toe "

Fig.4 shows experiment performance on data "Contraceptive Method Choice ", which has 3 classes. This data is difficult for classification, the best accuracy is around 52% reached by SVM. LSD is ranked second among all the models. It surpasses KNN5 and ALLIND when dictionary size is larger then 30% of training data. Meanwhile, LSD has smaller reconstruction error than SOMD and NGASD.

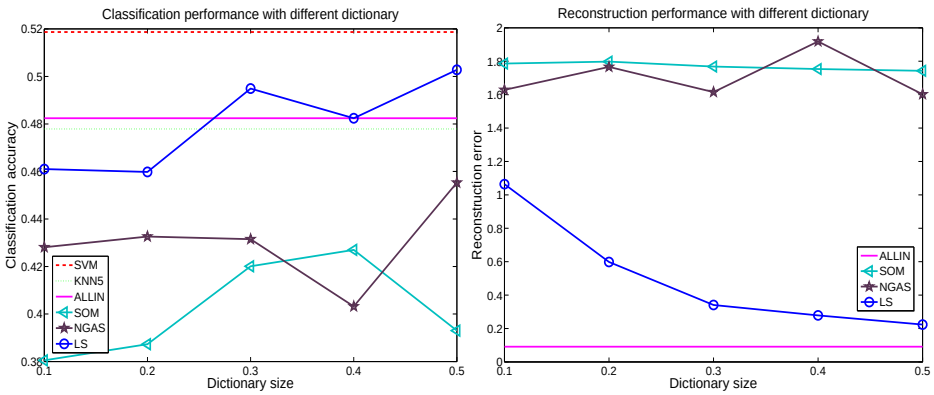

Fig. 4. Left : Classification comparison with data "Contraceptive Method Choice " **Right** : Reconstruction error with data "Contraceptive Method Choice "

Fig.5 shows the results on the data "Glass". In the left figure, ALLIND has the highest accuracy and LSD performs better than KNN5 and SVM. The performance of SOMD is also competitive with KNN5. It seems that SRC method has advantage than classic classifiers in this dataset. In the reconstruction view, LSD reach a stable level when the dictionary size is larger than 20% of training data.

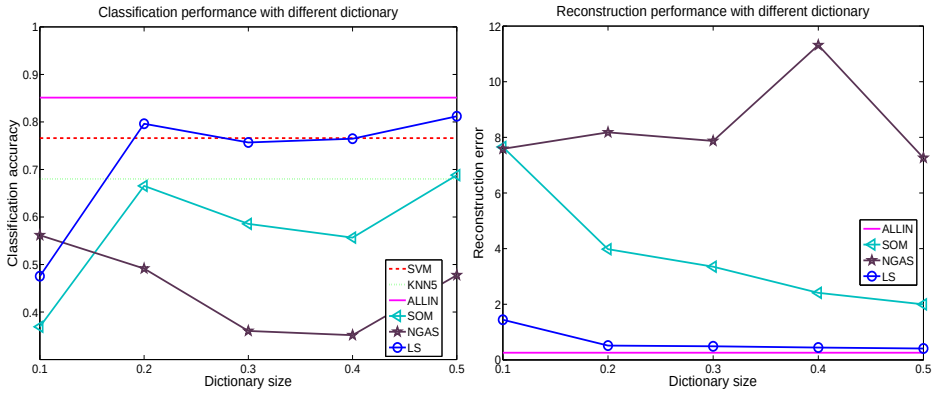

Fig. 5. Left : Classification comparison with data "Glass" **Right** : Reconstruction error with data "Glass"

The results of data "Image Segmentation" is shown in Fig.6. The performance of KNN5 are almost the same with ALLIN. With increasing the dictionary size,

the performance of LSD improves. And LSD reaches the level of SVM when the dictionary size is 40% of training data. In the right figure, the reconstruction error of LSD continuously decreases with more atoms are selected in the dictionary.

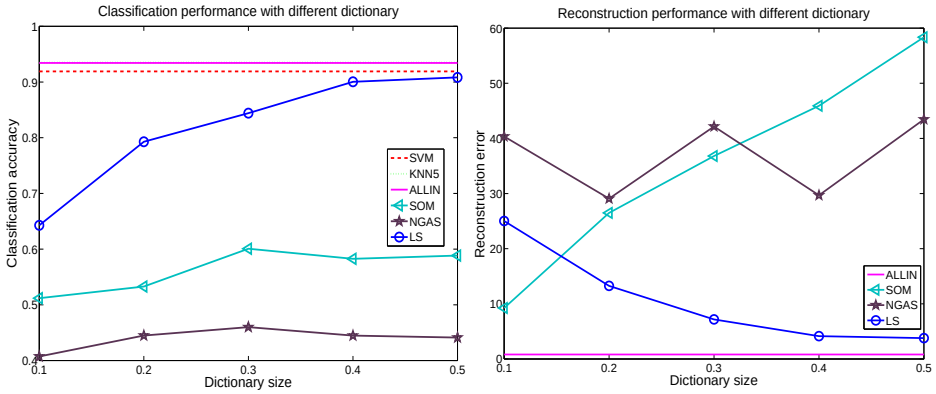

Fig. 6. Left : Classification comparison with data "Image Segmentation" **Right** : Reconstruction error with data "Image Segmentation"

Fig.7 shows the performance of data "Libras Movement". ALLIND has the best performance and LSD can be compared with SVM when the dictionary size is 40% of training data. In the reconstruction view, LSD has decreasing trend but NGAS has lower error.

Table 2. Accuracy [%] average in different dictionary sizes

Dictionary size (rate)	10%	20%	30%	40%	50%
SOMD	44.75	51.07	52.59	52.69	53.98
NGASD	43.28	42.65	39.55	38.40	44.89
LSD	56.78	68.10	70.30	73.85	75.85

From the results of different figures, we observe that LSD classification performance tend to be converged when the dictionary size in the range of 40%-50% training data. Then we get the average accuracy results based on different data in Talbe 2, and the average accuracy of SVM, KNN5 and ALLIND are 72.63%, 77.31% and 78.78% separately. It is interesting to know the accuracy of LSD with 40% training data can have better performance than SVM with 100% training data.

Overall, from the classification view, LSD is relatively better than SOMD and NGASD. The reason may due to the atoms labels in SOMD and NASD,

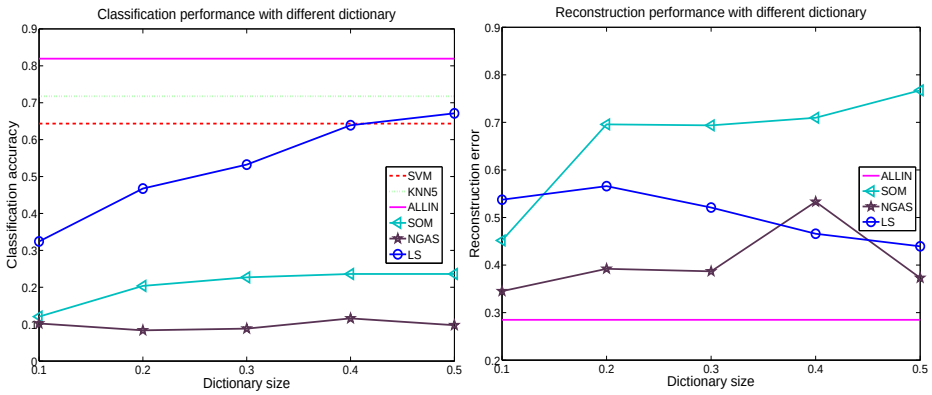

Fig. 7. Left : Classification comparison with data "Libras Movement " **Right** : Reconstruction error with data "Libras Movement "

which is assigned from training. While LSD, the atoms labels are associated with data which have unbiased information. However, in the construction view, we also can observe LSD has competitively smaller error compared with SOMD and NASD. It seems that unsupervised selection is more robust and meaningful than unsupervised transform for the dictionary learning.

5 Conclusion

We have presented a novel dictionary learning method which is inspired from geometry local information. It is an unsupervised learning to search the atoms in the training data. The experiments on the UCI data give a comprehensive study on different unsupervised dictionary learning models. The proposed LSD has shown effectiveness on SRC classification and reconstruction processes. LSD based on partial training information shows competitive performance with SVM based on full training information.

More experiments and theoretical analysis are of course need to assess the proposed dictionary in real application. Beyond this, there are some interesting points for future study: (1) Exploration of the connection between category data distribution and data geometry information in the dictionary learning; (2) Exploration of the relations between the unsupervised selection and unsupervised transform in the dictionary learning; (3) Exploration of the optimal dictionary size for dictionary learning.

5.1 Citations

For citations in the text please use square brackets and consecutive numbers: [1], [2], [4] – provided automatically by LaTeX's `\cite` ... `\bibitem` mechanism.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`

Appendix: Springer-Author Discount

LNCS authors are entitled to a 33.3% discount off all Springer publications. Before placing an order, the author should send an email, giving full details of his or her Springer publication, to `orders-HD-individuals@springer.com` to obtain a so-called token. This token is a number, which must be entered when placing an order via the Internet, in order to obtain the discount.

6 Checklist of Items to be Sent to Volume Editors

Here is a checklist of everything the volume editor requires from you:

- ☐ The final LaTeX source files
- ☐ A final PDF file
- ☐ A copyright form, signed by one author on behalf of all of the authors of the paper.
- ☐ A readme giving the name and email address of the corresponding author.