Towards IK-SVD: Dictionary Learning for Spatial Big Data via Incremental Atom Update

Lizhe Wang[1], Ke Lu[2], Peng Liu[1], Rajiv Ranjan[3], Lajiao Chen[1]

[1]Institute of Remote Sensing and Digital Earth, CAS, Beijing, 100094, China
[2]University of the Chinese Academy of Sciences, CAS, Beijing, 100049, China
[3]ICT, CSIRO, North Road Acton ACT 2601, Australia

There is a large group of dictionary learning algorithms that focus on adaptive sparse representation of data. Some of them are only applicable to small data, such as K-SVD and the non-parameter Bayesian method; other algorithms such as online dictionary leaning method (ODL) and recursive Least Squares dictionary learning method (RLS) could train the dictionary by using *relatively* large samples. However, almost all of them fix the number of atoms in iterations and use un-feasible schemes to update their atoms in the the dictionary learning process. Therefore, it is difficult for them to train a dictionary from big data, which is a very large data set with thousands of remote sensing images from long temporal sequences and form very large areas of the earth's surface. In this paper, we proposed a new dictionary learning algorithm by extending the classical K-SVD method. In the proposed method, when each new batch of samples data is added to the training process, we selectively introduce a number of new atoms into the dictionary. Furthermore, only a small group of new atoms as subspace controls the current orthogonal matching pursuit, construction of error matrix, and SVD decomposition process in every training cycle. The information from both old samples and new samples are explored in the proposed incremental K-SVD (IK-SVD) algorithm, but only the current atoms are adaptively updated, which makes the dictionary better represent all the samples without the influence of the redundant information from old samples. To represent the data set efficiently and sparsely, we also introduce a new constraint into the object function of atoms updating. Because both the sparse coding step and the atoms updating step are promoted, the proposed method is applicable to sparse the representation of very large spatial-temporal remote sensing image sets Some state-of-the-art algorithms are compared with the proposed method, and experiments confirm the better performance of the proposed method when dealing with spatial-temporal remote sensing big data.

***Index Terms*—Big Data, Sparse Representation, Dictionary Learning.**

I. Introduction

A. Motivating Scenario

BIG data is a collection of data sets so large and complex that it is difficult to deal with using traditional data processing algorithms and models. The challenges include acquisition, storage, search, sharing, transfer, analysis, visualization, etc. Scientists regularly encounter limitations due to large data sets in many areas, such as geosciences and remote sensing, complex physics simulations, and biological and environmental research. In remote sensing applications, the size of the data set grows in part because data are increasingly being gathered by many different satellite sensors with different resolutions and different spectral characters, and more importantly the data are often from long temporal image sequences and large area of the earth's surface.

Big data is difficult to deal with using traditional methods. How to represent a big data set is a fundamental problem in the research of big data, as most data processing tasks rely on an appropriate data representation. For many tasks, such as sampling, reconstruction, compression, retrieval, communication, classification, etc., a sparse data representation is preferred. And for remote sensing big data, the sparseness is more and more important for many algorithms such as image segmentation, image fusion, change detection, feature extraction, and image interpretation.

Manuscript received xxxxxx; revised xxxxxx. Corresponding author: Lizhe Wang (email: lzwang@ceode.ac.cn).

B. Related Work

There is a long research history on how to sparsely represent a signal or data by a set of basis. We also call this set of basis a dictionary. There are at least two different classes of dictionary: one is an analytic dictionary; the other is an un-analytic dictionary.

Many of the earlier research studies on sparse representation focus on analytic dictionaries. For example, the Fourier dictionary is designed for smooth functions, while the wavelet dictionary is designed for piecewise-smooth functions with point singularities. Recently, the idea of curvelet transform was proposed by Donoho in [1]. Each curvelet atom is associated with a specific location, orientation, and scale, which make it efficiently represent the smooth curves. The bandelet transform [2] represents one of the most recent contributions in the area of signal-adaptive transforms. There are some other adaptive analytic dictionaries, such as directionlet transform [3] and grouplet transform [4], which are also popular in sparse representation research areas.

Another large class of dictionary is un-analytical. Unlike decompositions based on a predefined analytic base (such as wavelet) and its variants, we can also learn an overcomplete dictionary without analytic form, which neither have fixed forms of atoms nor require that the basis vectors be orthogonal. The basic assumption behind the learning approach is that the structure of complex incoherent characters can be more accurately extracted directly from the data than by using a

mathematical description. The method of optimal directions (MOD) [5] is one of the earliest un-analytical methods. In generalized PCA (GPCA) [6], each sample is represented by only one of the subspaces. K-SVD [7] focuses on the same sparsification problem as the MOD, and employs a similar block-relaxation approach. The main contribution of the K-SVD is its atoms updating method. Rather than using a matrix inversion, it performed atom by atom in a simple and efficient process. Nonparametric Bayesian dictionary learning [8], employs a truncated beta-Bernoulli process to infer an appropriate dictionary, and obtains significant improvements in image recovery [8].

C. The Research Problem

Many recent algorithms for un-analytic dictionary learning are iterative batch procedures. They access the whole training set at each iteration and minimize the cost function under some constraints. These algorithms cannot efficiently deal with very large data sets, or dynamic data changing over time. To address these issues, a popular method online dictionary learning method [9] (ODL) was proposed. Another competitive methods is the recursive least squares dictionary learning (RLSDL) algorithm [10]. Both OLD and RLSDL have the ability to train large sets, which in theory are not a dictionary method specialized to the particular (small) training set. However, they may encounter some problems while dealing with real remote sensing big data. First, they update all atoms for every new sample, which may be unrealistic when there are so many atoms for a real big data set; second, the fixed the number of the atoms in the dictionary learning process is not very adaptable. For the dictionary learning of a big data set, how many atoms there are and which atoms need to be updated should be dynamic.

D. Contributions

This paper proposes the incremental K-SVD (IK-SVD) algorithm, which yields dynamic dictionaries by sequentially updating the dictionary atoms one part every time. Furthermore, when the number of atoms changes with the training process, the dictionary will be able to represent the spatial-temporal remote sensing big data efficiently and sparsely.

II. Dictionary Learning for Finite Training Data Sets

Classical dictionary learning techniques for sparse representation considering a data sample $y \in R^h$ can be described as $y = D\alpha$, where $D \in R^{h \times n}$ is a dictionary with n atoms, and $\alpha \in R^n$ is the coefficients for the sparse representation. We typically consider the case $n > h$, suggesting that the dictionary is redundant. The number of non-zeros coefficients in the representation is denoted as $k = \|\alpha\|_0$, where k is expected to be very small. $y = D\alpha$ implies that the sample y can be characterized as a linear combination of a few columns from the dictionary $D \in R^{h \times n}$, which is also referred as to the set of atoms. Then, the problem is

$$\min_{D,\alpha} \|y - D\alpha\|_2^2 \quad subject \quad to \quad \|\alpha\|_0 \leq k, \tag{1}$$

TABLE I: Notation

symbol	meaning
y	sample data
$Y = \{y_1, \cdots, y_r\}$	small data set
$\{Y_1, \cdots, Y_s\}$,	large data set
α	coefficient vector
$\alpha_i = \{\alpha_i(1), \cdots, \alpha_i(n)\}$	α_i with n components
$X = \{\alpha_1, \cdots, \alpha_r\}$	coefficients set
$\{X_1, \cdots, X_s\}$,	large coefficients set
α_T^j	the jth row in X
d	atom
$D = \{d_1, \cdots, d_r\}$	atoms set
λ	regularization parameter
$\omega_k = \{i \| 1 \leq i \leq r, \alpha_T^k(i) \neq 0\}$	no-zero support of α
E	error matrix

In another expression of the object function and constrains, we can also control the error of the reconstruction as $\|y - D\alpha\|_2^2 \leq \sigma$. Usually, there is a group of samples that need to be represented, so it is denoted as $Y = \{y_1, \cdots, y_r\}$, and the coefficients set is denoted as $X = \{\alpha_1, \cdots, \alpha_r\}$, where $X \in R^{n \times r}$. Now, the sparse representation and dictionary learning problem is defined as

$$\min_{D,\alpha_1,\cdots,\alpha_r} \sum_{i=1}^{r} \|y_i - D\alpha_i\|_2^2 + \lambda \sum_{i=1}^{r} \|\alpha_i\|_0, \tag{2}$$

It is equal to

$$\min_{D,X} \|Y - DX\|_2^2 + \lambda \|X\|_0, \tag{3}$$

where $\|\cdot\|_2$ is L_2 norm, $\|\cdot\|_0$ is L_0 norm, and λ is a regularization parameter. It is well known that L_0 regularization yields a sparse solution for X, which was proved in reference [11]. This problem to search X is also known as the Lasso or basis pursuit. To prevent the dictionary D from having arbitrarily large values (which would lead to arbitrarily small values of α), it is common to constrain its atoms $D = \{d_1, \cdots, d_n\}$ to have an L_2-norm less than or equal to one [7]. The problem of minimizing the object function (3) is a joint optimization problem with respect to the dictionary D and the coefficients X in the sparse decompositions. The object function is not jointly convex, but it is convex with respect to each of the two variables D and X when the other one is fixed.

There is a large group of research [7] [9] [10] [8] that focuses on how to find a good dictionary by training samples and, at the same time, how to represent a small data set $Y = \{y_1, \cdots, y_r\}$ sparsely. Under most circumstances, there are two stages in the dictionary leaning algorithms: one is the sparse coding stage, which searches for the optimal solution of equation (4),

$$\min_{\alpha} \|Y - D\alpha\|_2^2 + \lambda \|\alpha\|_0. \tag{4}$$

With a fixed dictionary D, the sparse coding problem of equation (4) is an L_0 regularized problem that also could be

substituted by an L_1 problem. It can be solved using many methods [12] [13]. The other is the atoms updating stage, which means finding the solution of equation (5).

$$\min_{D} \|Y - D\alpha\|_2^2. \tag{5}$$

The atoms updating stage in equation (5) is the main characteristic that distinguish many different popular dictionary learning methods. The OLD and RLSDL methods employ adynamic schemes in the atom updating stage, so that they can train an infinite data set in theory. The classical K-SVD method has very good performance on small data, but it is impossible to perform K-SVD dictionary learning for big data because we could neither read all samples data into the computer memory nor can we perform SVD decomposition of a very large matrix.

However, K-SVD has its own advantages. Its dictionary update scheme is a good model with clear mathematics and physics meanings. To sparsely represent the big data set from remote sensing, we will extend the K-SVD algorithm and explore the redundant features of a spatial-temporal remote sensing image set. In the next section, we propose the incremental K-SVD algorithm, which can introduce new atoms into the dictionary and update the small part of the dictionary by using only the current sample set step by step in each iteration. We will discuss in detail how to train a dictionary from a large spatial-temporal data set by the proposed incremental K-SVD (IK-SVD).

III. Dictionary Learning by IK-SVD

In the last section, we analyzed the classical K-SVD method, which is applicable to small data sets. Now assume that there is a big data set $\{Y_1, \cdots, Y_s\}$, where s means a different time or a different location. These multi-spatial-temporal data share similar features and difference from each other. Since redundancy information always exists in a large spatial-temporal data set $\{Y_1, \cdots, Y_s\}$, it is possible to represent it sparsely by dictionary leaning.

Most of traditional methods are not applicable to every large data. This means that we cannot train all the samples in a big data set $\{Y_1, \cdots, Y_s\}$ at one time to get the final dictionary D. In a learning algorithm for a big data set, both the number of the atoms and the samples to be used should change dynamically. Therefore, the problem becomes: If for the data set $\{Y_1, \cdots, Y_s\}$ we have already obtained its dictionary $D_s = \{d_1, \cdots, d_n\}$, then for the next scene of image Y_{s+1} with the index $s+1$, we need to find a new dictionary $D_{s+1} = \{_1, \cdots, d_n, d_{n+1}, \cdots, d_{n+m}\}$, which has m more atoms as $d_{n+1}, \cdots, d_{n+m}$ added and is able to represent the data set $\{Y_1, \cdots, Y_{s+1}\}$ sparsely.

Obviously, we hope that the latest atoms $d_1, \cdots, d_n$ are still reserved in the new dictionary $\{d_1, \cdots, d_n, d_{n+1}, \cdots, d_{n+m}\}$. We also hope that, when every new subset data Y_{s+1} is trained, only a few new atoms are reasonably added to D_{s+1}, so that we can efficiently and sparsely represent both $\{Y_1, \cdots, Y_s\}$ and Y_{s+1} . As a result, in the training process, we could obtain $\{X_1, \cdots, X_s, X_{s+1}\}$, which is the sparse coefficients matrix sequence for the data set $\{Y_1, \cdots, Y_s, Y_{s+1}\}$. Since we already defined that D_s is a part of D_{s+1} and D_s can already sparsely represent $\{Y_1, \cdots, Y_s\}$, we only need to update coefficients X_{s+1}, which is relate to the sparse representation for Y_{s+1} based on dictionary D_{s+1}. Now we define the new object function for the incremental learning model as

$$\min_{D_{s+1}, X_{s+1}} \|Y_{s+1} - D_{s+1}X_{s+1}\|_2^2 + \lambda\|X_{s+1}\|_0, \tag{6}$$

When training data $Y_1, \cdots, Y_s, Y_{s+1}$, we assume that they have the same number of r samples, and then we have

$$Y_{s+1} = \{y_1, \cdots, y_r\} \tag{7}$$

Since there are r samples in each Y_{s+1}, the corresponding coefficient X_{s+1} is

$$X_{s+1} = \{\alpha_1, \cdots, \alpha_r\}. \tag{8}$$

In equation (8), the coefficient vector α_i with more components, where $1 \leq i \leq r$, becomes

$$\alpha_i = \{\alpha_i(1), \cdots, \alpha_i(n), \alpha_i(n+1), \cdots, \alpha_i(n+m)\} \tag{9}$$

It can be observed that there are more components, such as $\alpha_i(n+1), \cdots, \alpha_i(n+m)$, in the coefficient vector α_i in equation (9), because there are more atoms in the current dictionary D_{s+1}. Since we cannot train all the samples at one time, we construct and update every small group of the atoms for every new training set Y_{s+1}.

Actually, we care more about the current atoms $d_{n+1}, \cdots, d_{n+m}$ and their coefficients. In an extreme case, for the current Y_{s+1}, if $\alpha_i(1), \cdots, \alpha_i(n)$ in every coefficient vector α_i are efficient and sparse enough, even $d_{n+1}, \cdots, d_{n+m}$ are not necessary and $\alpha_i(n+1), \cdots, \alpha_i(n+m)$ can all be zeros. However, usually there are new atoms, such as $d_{n+1}, \cdots, d_{n+m}$, that need to be added and updated because for a large spatial-temporal data set because there are always some image features in Y_{s+1} thant cannot be efficiently represented by atoms trained from $\{Y_1, \cdots, Y_s\}$.

When we solve equation (6), following the idea of classical K-SVD, the jth row in X_{s+1}, is denoted as α_T^j (this is not the vector α^j, which is the jth column in X). For an arbitrary new kth atom, the first term of the object function in equation (6) can be denoted as

$$\begin{aligned} &\|Y_{s+1} - D_{s+1}X_{s+1}\|_2^2 = \\ &\|Y_{s+1} - \sum_{j=1}^{n} d_j\alpha_T^j - \sum_{j=n+1}^{k-1} d_j\alpha_T^j - \sum_{j=k+1}^{n+m} d_j\alpha_T^j - d_k\alpha_T^k\|_2^2, \end{aligned} \tag{10}$$

which is the changing form of the object function of the proposed incremental dictionary learning. In equation (10), there are two obvious differences from the classical K-SVD model: the first difference is that the current sample Y_{s+1} and the old atoms $d_1, \cdots, d_n$ trained by old samples are linked and combined into one object function; the second one is $n+1 \leq k \leq n+m$, which means that for the new training

samples Y_{s+1}, we will only update new the atoms within $d_{n+1}, \cdots, d_{n+m}$.

Therefore, the equation should be rewritten as

$$\|Y_{s+1} - D_{s+1}X_{s+1}\|_2^2 = \|E_{s+1}^k - d_k\alpha_T^k\|_2^2 \tag{11}$$

where

$$E_{s+1}^k = Y_{s+1} - \sum_{j=1}^{n} d_j\alpha_T^j - \sum_{j=n+1}^{k-1} d_j\alpha_T^j - \sum_{j=k+1}^{n+m} d_j\alpha_T^j \tag{12}$$

We have decomposed the multiplication $D_{s+1}X_{s+1}$ into the sum of $n+m$ matrices. Among those $n+m$, $n+m-1$ terms are assumed fixed, and one (the kth) remains in question. However, it is different from the traditional K-SVD method, that for the new training sample data Y_{s+1}, we will never update atoms of $d_1, \cdots, d_n$ that are already trained by $\{Y_1, \cdots, Y_s\}$. Every time, what will be updated are only atoms of $d_{n+1}, \cdots, d_{n+m}$. Therefore, for Y_{s+1}, we only calculate the current error matrix E_{s+1}^k, where $n+1 \leq k \leq n+m$. The meaning of E_{s+1}^k is different form that of reference [7]. The proposed matrix E_{s+1}^k stands for the error for the current samples Y_{s+1} but not all history samples, when atoms $d_1, \cdots, d_n$ are fixed and when the atom d_k is removed, where $n+1 \leq k \leq n+m$. Respectively, we also need to define a new ω_k for new atoms as

$$\omega_k = \{i | 1 \leq i \leq r, \alpha_T^k(i) \neq 0\} \tag{13}$$

where $n+1 \leq k \leq n+m$.

Note that the error matrix E_{s+1}^k stands for how well the dictionary D_{s+1} without d_k can represent the current training data Y_{s+1}, and the information from known atoms $d_1, \cdots, d_n$ is still associated with E_{s+1}^k. On the other hand, initial value of the D_{s+1} for the proposed method is also different from that of traditional K-SVD, which will be discussed in the next section.

IV. Estimate the initial value of the new Atoms the Dictionary

Although we can add new atoms to the current dictionary, it is still very difficult to set the initial value of the new atoms when a batch of new samples is introduced into the training process. If the old dictionary D_s can efficiently and sparsely represent the new samples Y_{s+1}, we do not need to create new atoms and put them into D_{s+1}. However, there are often new image features from the new samples, which cannot be efficiently represented by old atoms, so we often need to add new atoms. We will select special samples as the initial value of the new atoms. If we set improper initial values for the new atoms, the training process will be slow and inefficient. Therefore, a good choice of new atoms is very important to the incremental dictionary learning.

When each new Y_{s+1} is considered, we first perform a sparse coding for Y_{s+1} using dictionary D_s to evaluate how well the old dictionary D_s could represent the current samples Y_{s+1}. Then, there is

$$\min_{\bar{X}_s} \|Y_{s+1} - D_s\bar{X}_s\|_2^2 + \lambda\|\bar{X}_s\|_0 \tag{14}$$

We call equation (14) as the initial representation. For this initial representation, for an arbitrary coefficient α_i vector within $\bar{X}_s$, it has n components as equation (15) but not $n+m$ components.

$$\alpha_i = \{\alpha_i(1), \cdots, \alpha_i(n)\} \tag{15}$$

The information from coefficient α_i characterizes the relationship between new samples Y_{s+1} and the old atoms D_s. To utilize the sparse coefficients to assist in introducing new atoms, we use the idea of active learning [14] to set the initial value for new atoms. The basic idea of active learning is to iteratively enlarge the training set by requesting an expert to label new samples from the unlabeled set in each iteration [15]. In this paper, we propose to use the entropy of information theory to decide which new samples will be the initial value of new atoms.

First, we select the samples from Y_{s+1} whose coefficients are not sparse enough when we sovle equation (14). Then, among all the samples that cannot be sparsely represented by old atoms, we need to select the samples showing maximal disagreement between the different atoms, and they will be the initial value of the new atom. It is a little similar to active learning considered an MI-based criterion [14]. The difference is that we do not label the new sample but set it as a new atom. Now, we define the new atom d_{new} as

$$d_{new} = \max_{\alpha_i \in X_{s+1}} H(\alpha_i) \tag{16}$$

where

$$H(\alpha_i) = \sum_{j=1}^{n} p(l = d_j|\alpha_i) log(p(l = d_j|\alpha_i)) \tag{17}$$

where $p(l = d_j|\alpha_i)$ is defined by normalized sparse coefficients as

$$p(l = d_j|\alpha_i) = \frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)} \tag{18}$$

Actually, $H(\alpha_i)$ also is

$$H(\alpha_i) = \sum_{j=1}^{n} \frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)} log(\frac{\alpha_i(j)}{\sum_{b=1}^{n} \alpha_i(b)}) \tag{19}$$

In the iteration of the proposed method, we will first select a group of samples that cannot be sparsely represented by the old dictionary, and then we calculate their entropy by equation (17) and select m samples with the largest entropy as the initial value of the new atoms.

Now, we sum up the proposed dictionary learning algorithm as follows:

1. The big data set is $\{Y_1, \cdots, Y_s, \cdots, Y_S\}$, where $1 \leq s \leq S$. Train the sample subset Y_1 by classical K-SVD, and get the initial dictionary $D_1 = \{d_1, \cdots, d_n\}$. Set s=2, and J=1.

2. Solve object function (14), select m samples based on equation (16), and $D_s^{(J)} = D_{s-1} \bigcup \{d_{n+1}, \cdots, d_{n+m}\}$.

3. **Sparse coding stage**: use the OMP algorithm to compute the representation Y_s by the solution of

$$\min_{X_s} \|Y_s - D_s^{(J)} X_s\|_2^2 + \lambda \|X_s\|_0$$

4. **Atoms update stage**: for each new atom d_k in dictionary $D_s^{(J)}$, where $k = n+1, \cdots, n+m$, update it as follows:

–Define the group of examples that use this atom d_k, $\omega_k = \{i | 1 \leq i \leq r, \alpha_T^k(i) \neq 0\}$. Compute the overall representation error matrix by equation (11), and get E_s^k.

–Construct $\hat{E}_s^k$ by choosing only the columns corresponding to ω_k within E_s^k. Apply SVD decomposition $\hat{E}_s^k = U \Lambda V^T$; choose the first column of U to be the updated atom d_k. Update the coefficient vector α_T^j to be the first column of V multiplied $\Lambda(1,1)$.

5. Update the dictionary.

6. Set J=J+1. Repeat step 3 through 6 until convergence.

7. s=s+1, n=n+m, and J=1. Repeat step 2 through 7, until all the data in $\{Y_1, \cdots, Y_S\}$ are trained.

The word convergence means that, in the inter-loop, the error satisfies $\|Y_s - D_s^{(J)} X_s\|_2^2 \leq \sigma$ or the sparsity satisfies $\|X_s\|_0 \leq k$. Fig. 1 is the flow chart of the algorithm.

Now, we can find the differences between the proposed IK-SVD, ODL and RLSDL. For IK-SVD, we always update the atoms based on the new sample data that cannot be well represented by old dictionary. The number of the atoms for IK-SVD is changing and increasing in the training process, which make IK-SVD very flexible. The atoms update stage of ODL is similar to gradient descendent, therefore the most important is to find a appropriate gradient that fits both new sample data and old sample data. The RLSDL algorithm is same as the recursive least squares algorithm for adaptive filtering. Thus, a forgetting factor is very important to the atoms update stage of RLSDL.

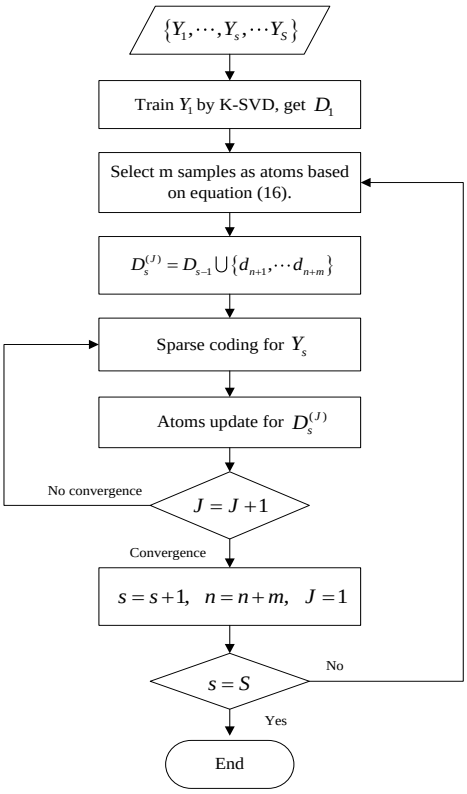


Fig. 1: The flow chart of the algorithm.

V. Experiments and Results

In the experiments, we use the image data set of the Landsat satellite, who represents the world's longest continuously acquired collection of space-based moderate-resolution land remote sensing data. In the past four decades, the imagery data set has provided a unique and extremely rich resource for the research on agriculture, geology, forestry, regional planning, education, mapping, and global change.

Since July 23, 1972, there has been a series of Landsat satellites missions, from Landsat-1 to Landsat-8. In our experiments, different Landsat satellites data sets are included, which have different resolution and different spectral characteristics. Because the series of Landsat satellites continuously acquired image data for four decades, the whole image data volume is large enough to be big data. In general, it is hard to precisely model remote sensing big data. In the following experiments, we train the samples by randomly selecting some subsets of the big data.

The proposed algorithm of incremental K-SVD (IK-SVD) is compared with two other dictionary learning algorithms ODL [9] and RLSDL [10]. The volume of the global landsat data is so large that it is unrealistic to put all of them into the dictionary learning experiment process for the three algorithms. However, because of the highly redundant nature of the massive temporal-spatial remote sensing image set, we could compare the three methods by the random selection of the sample data in the big data set. We mainly compare the performance of the algorithms in two respects: one is the precision of the reconstruction; the other is the sparse extent of the decomposition. The two characters are mutual restrictive. Therefore, we compare one feature while the other feature is fixed. For convenience of comparison, we use the OMP method for the reconstruction for all of the three methods. This means that, we train the dictionary using different models but solve the sparse coefficients using the same reconstruction algorithm.

A subset of the Landsat global image data set is selected for the validations. The subset of data used in these experiments are from the 2008 and 2009 whole year data, which cover the whole area of China more than 9000,000 square kilometers. In this data subset, only one scene multispectral image set is selected for every location of earth's surface. Some image data that were destroyed or with too much clouds are not included in the test data set. The volume of the data set with all bands is about 650 GB. We train the dictionary by randomly selecting 30 GB data samples within the data set, and we validate the performance of the three algorithms by randomly selecting another 10 GB data samples for each test.

Remote sensing big data from Landsat satellites contain many long time sequence data sets for many location of the earth's surface. The data subset for the Beijing area, in northen

China, which covers 16,411 square kilometers, was selected for the tests. The time ranges from 1983 to 2013, and some of data with too much cloud were removed from the data set. There are many forests, cities, and mountains in this area, which makes the texture information very rich. Furthermore, the high degree of climatic seasonality is another characteristic of the data subset. The volume of the tested data subset is about 110 GB. We train the dictionary by randomly selecting 3 GB data samples within the data set, and we validate the performance of the three algorithms by randomly selecting another 5 GB data samples for each different test parameter. In our experiments, we set the size of the incremental data as 8x8x100B for each iteration for all the three method.

Fig. 2, shows the dictionaries trained by different methods. Fig. 2(a) shows the atoms trained by the proposed IK-SVD, Fig. 2 (b) shows the atoms trained by RLSDL, and Fig. 2(c) shows the atoms trained by ODL. Actually, the number of atoms trained ranges from 150 to 3000. However, we cannot show them completely in each figure. Only a very small part of the atoms for the different algorithms is shown in Fig. 2(a-c). It is impossible to precisely judge whose performance is better based on what the atoms look like. However, we could infer that they have different effectiveness and adaptiveness for different textures of large data sets. Furthermore, we can observe that some atoms contain too much noise in RLSDL, and some atoms are overly smooth and contain very few textures in ODL. However, the texture in the IK-SVD atoms looks richer than those of the other two dictionaries. Unlike some predefined atoms of the analytic dictionary, the atoms in the adaptively learned un-analytic dictionary change with the sparsity and presentation precision. We find some difference between atoms with different representation precision. In Fig. 2, on the left, the error $\sigma = 10$, and on the right $\sigma = 20$. Therefore, we can observe that the features in the right column in Fig. 2 are smoother than those of the left column.

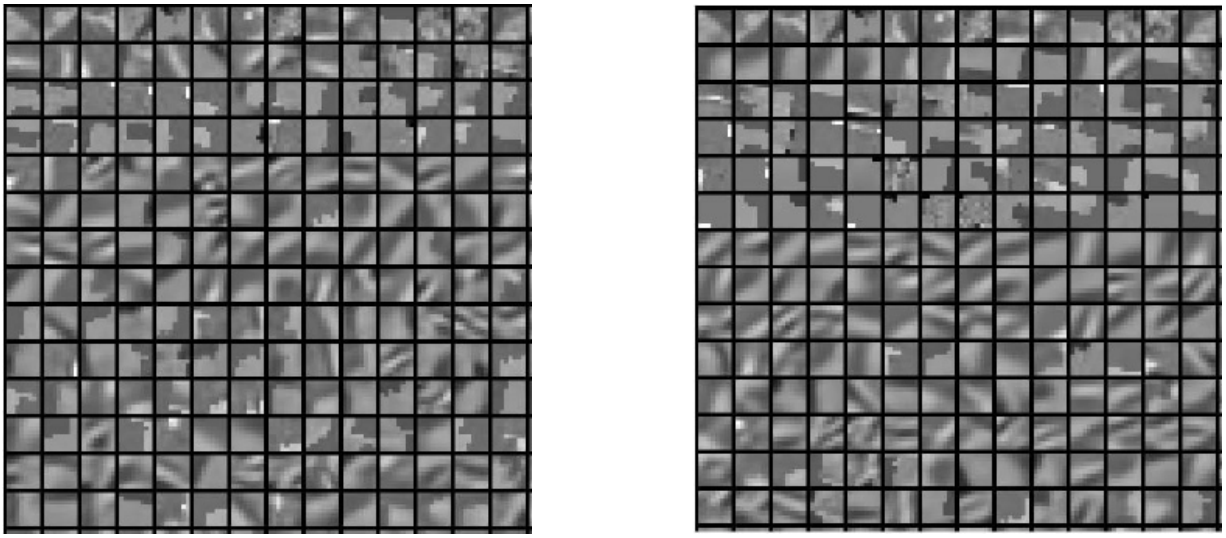
In Fig. 3, the precision of the reconstruction by different methods is compared. When comparing reconstruction errors in Fig. 3, for an arbitrary image data subset $Y_s = \{y_1, \cdots, y_r\}$ the error is defined as

$$E = \sum_{i=1}^{r} \|y_i - D\alpha_i\|_2^2. \tag{20}$$

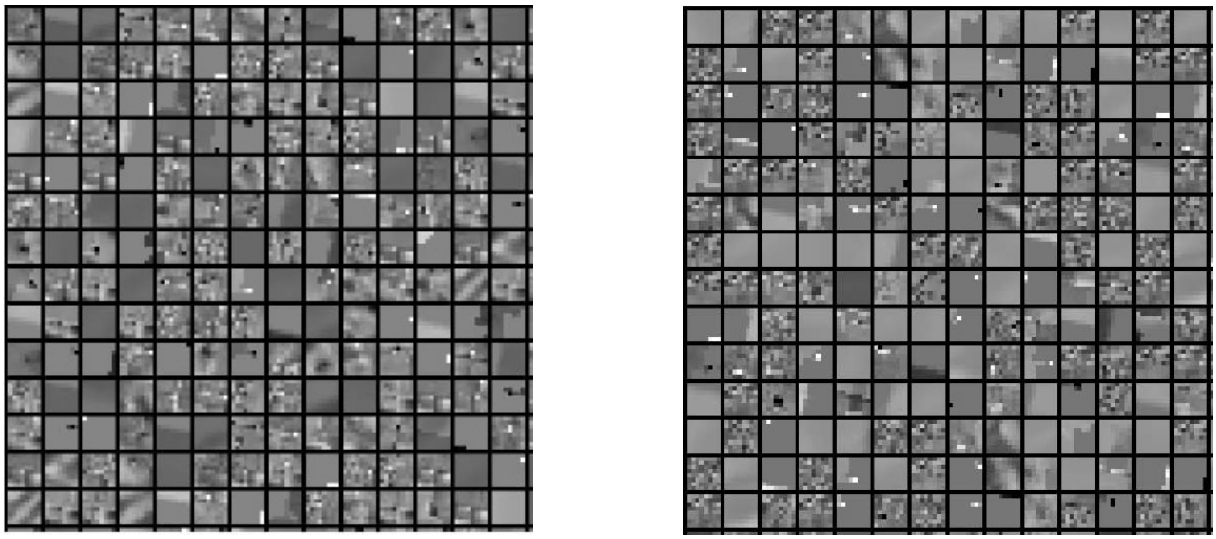
In this experiments, while training the atoms, we control the sparsity of the decomposition for each algorithm and compare the errors E in equation (20) for different methods. Therefore, it is the constrain optimal problem as shown in equation (1). To better validate the performance of the method, IK-SVD, ODL and RLSDL, while training the dictionary, we set the different sparsity of the representation at 5%, 10%, and 15%. For ODL and RLSDL, it is easy for the training to set the sparsity. But for the proposed IK-SVD, because we dynamically introduce the new atoms for the model, we need to set the threshold for when we should add some new atoms. For a set of samples Y_{s+1}, while we use OMP to sparsely decompose the samples to meet the controlled sparsity k, and if the PSNR (peak signal-to-noise ratio) for the sparse coding stage is smaller than 34 dB, we will introduce new atoms into the dictionary. For real big data with an un-limitation number of images, its impossible to take all the data as the samples. We randomly select sample data from the two data sets mentioned above and experiment on both long time sequence data and large area data. While we control the sparsity of the coefficients at 5% in Fig. 3(a), 10% in Fig. 3(b) and 15% in Fig. 3(c), we can observe that the precision of the proposed method roughly between those of ODL and RLSDL. Most IK-SVD results have higher precision than RLSDL but are very close to those of ODL, although we select different samples and set different sparsity.

In Fig. 4, the sparsity of the coefficients of the different methods is compared. In this experiments, we control the error or PSNR of the sparse decomposition for each algorithm. While the reconstruction error is fixed, the sparsity for the coefficients of every sample image is different for the three method. In order to comprehensively validating the performance of the algorithms, we also test the sparsity while reconstruction error $\sigma = 10$, $\sigma = 15$, and $\sigma = 20$. Unlike what is shown in Fig. 3, we experiment separately on long time sequence data and large area data. Therefore, the left side of Fig. 4(a-c) shows the results of data subset of Beijing area ranging for 1983 to 2013, and the right side of Fig. 4(a-c) shows the results of the data subset from the 2008 and 2009 whole year data, which cover the whole China area. We can observe that, for both long time sequence data and large area data, the number of nonzero coefficients of the proposed IK-SVD is less than those of ODL and RLSDL. Furthermore, the better sparsity feature of IK-SVD is relatively steady but not obviously affected by the representation precision σ.

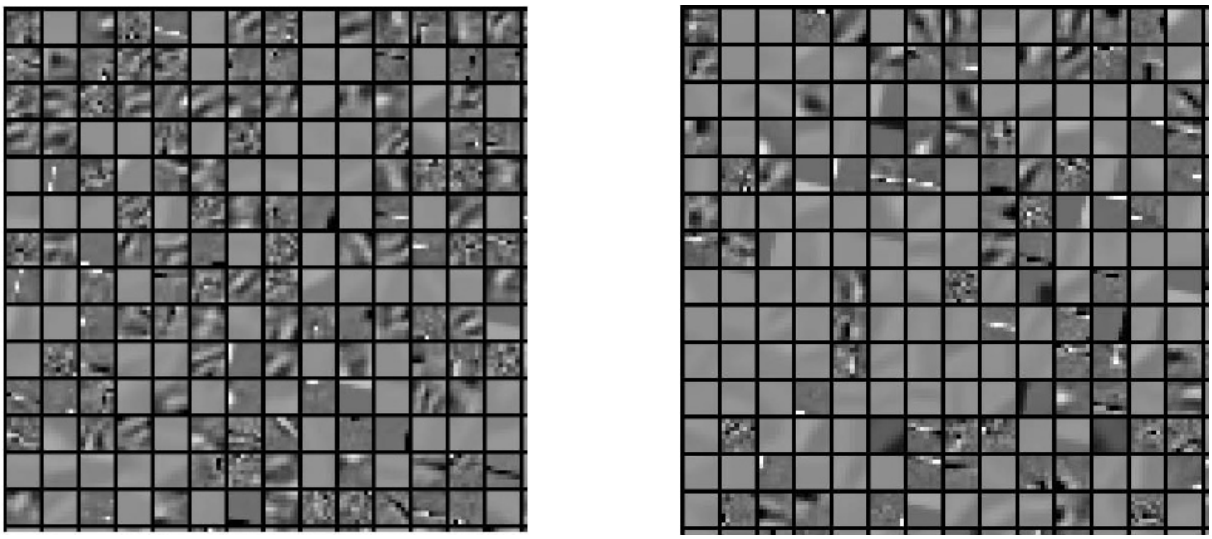
We also compare the time consumption of the three methods in Table II. It is hard to very fairly compare the speed of the three methods because their different program styles, data structures, and IO scheme. However, since the original intention of ODL and RLSDL was to design an algorithm competent to deal with unlimited large data sets, it is worth testing the algorithms using a large data set. If we train a large number of atoms, it should consume more time than a small number. In Table II, based on our experiments, when the number of atoms for RLSDL is larger than 500, the time consumption of the algorithm is too long to accept. Therefore, for BJ (Beijing area) data, we have to fix the number of atoms for RLSDL as 150 to make its time consumption acceptable. For time consumption, the representation precision is also an important factor in the dictionary learning. Very low errors make the proposed method create more atoms, and it makes the training slow. In addition, the controlling of the sparsity also affected the speed of the training. The more the sparse constraints controlled, the less time is used. We also find that the performance of RLSDL is not steady, as the time consumption dramatically increases and becomes unacceptable with the increasing number of the atoms. For the same error or sparsity, ODL is very fast when the number of the atoms is small, but when the number of the atoms exceeds 2600, ODL is slower than IK-SVD. The fixed number of atoms makes ODL obviously faster with the decreasing of the atoms. However, for IK-SVD there are more atoms that join the new training process. This makes the acceleration of IK-SVD not as obviously as ODL when the number of atoms decreases.



(a) IK-SVD

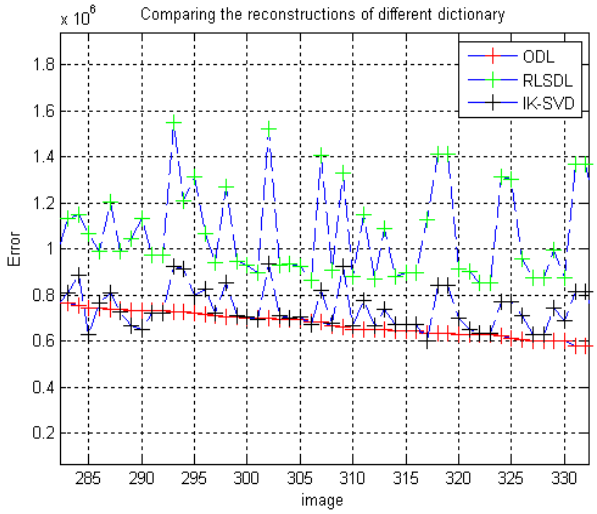


(b) RLSDL

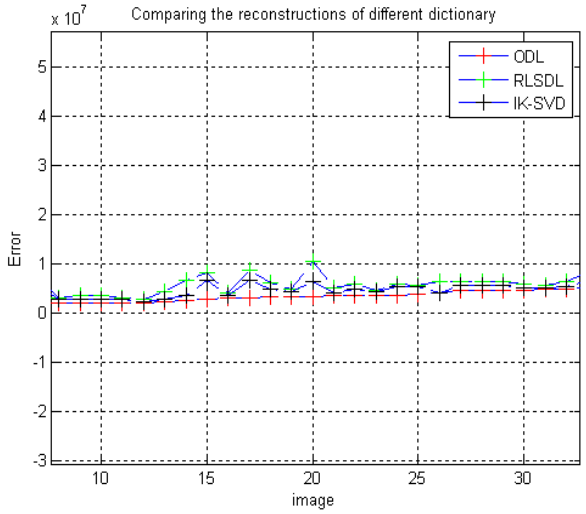


(c) ODL

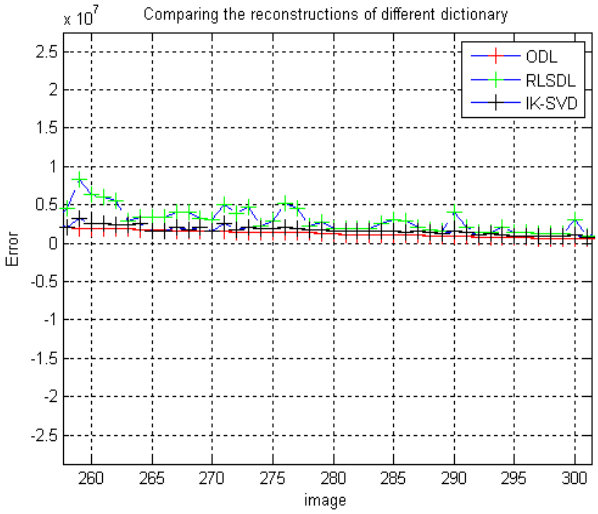
Fig. 2: Different dictionaries trained by sample data. The atoms in the left column are with constraint parameter $\sigma = 10$, while the atoms in the right column are $\sigma = 20$. We can observe that some atoms contain too much noise in RLSDL, and some atoms are overly smooth and contain very few textures in ODL. However, the texture in the IK-SVD atoms looks richer than those of the other two dictionaries.



(a) sparsity is 5%

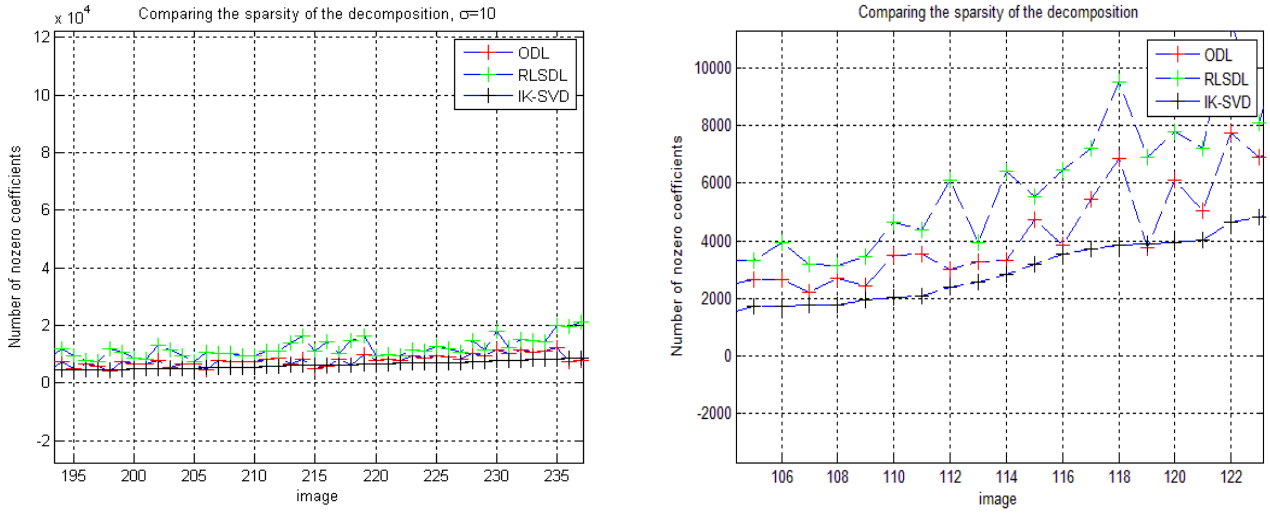


(b) sparsity is 10%

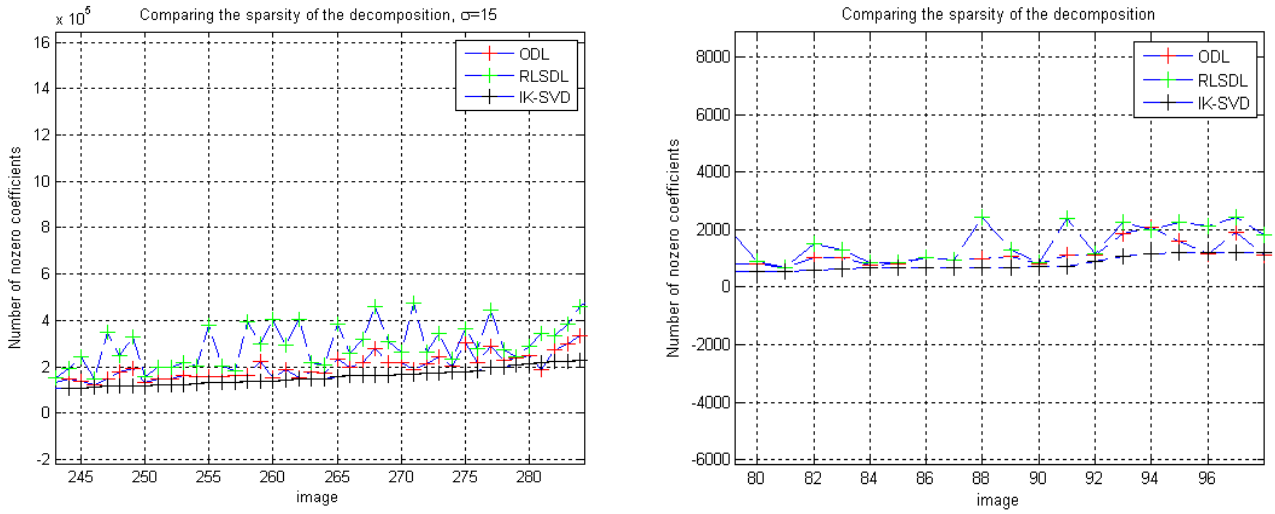


(c) sparsity is 15%

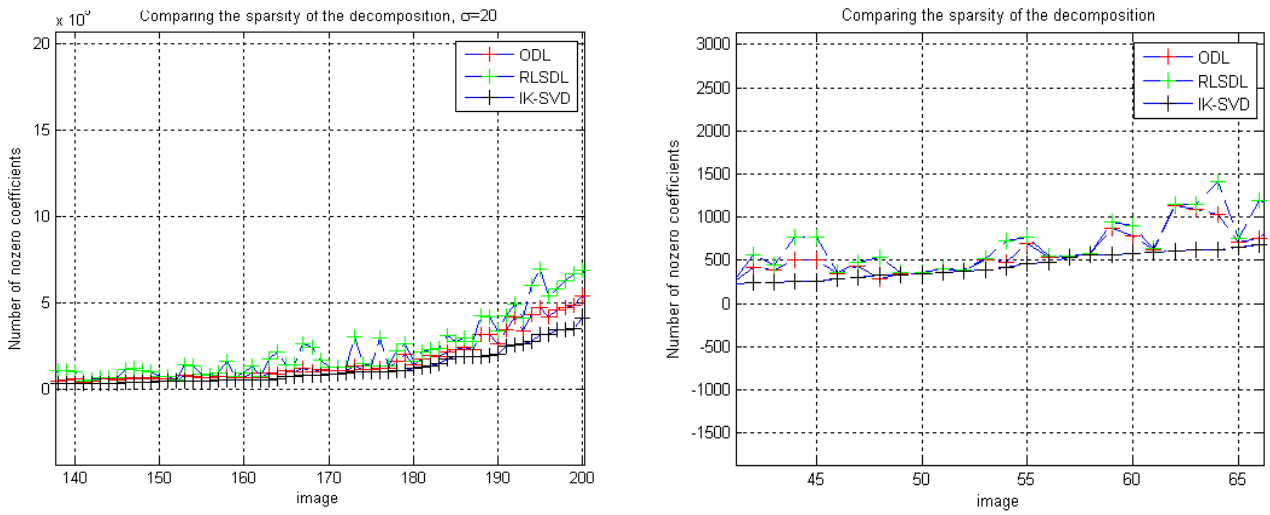
Fig. 3: Precision of the reconstruction while controlling the sparsity.



(a) $\sigma = 10$



(b) $\sigma = 15$



(c) $\sigma = 20$

Fig. 4: Sparsity of different methods while controlling the error.

TABLE II: The number of atoms and the time of the training for different methods and different data sets. “h” means hour. “BJ” means data set from Beijing area, and “CH” means data set from China area.

			Sparsity			Error		
Data	Method		5%	10%	15%	10	15	20
BJ	IK-SVD	atoms	895	205	175	2,610	1,260	410
		time	**1.7h**	2.8h	3.5h	**10h**	4h	1.5h
	ODL	atoms	895	205	175	2,610	1,260	410
		time	2.5h	1.3h	0.7h	13.3h	2.8h	1.0h
	RLSDL	atoms	150	150	150	150	150	150
		time	3.3h	4.2h	4.5h	4.1h	1.9h	0.8h
CH	IK-SVD	atoms	2,140	805	505	2,730	1,640	770
		time	10.2h	5.9h	5.1h	**11.4h**	5.5h	1.75h
	ODL	atoms	2,140	805	505	2,730	1,640	770
		time	**9.6h**	2.3h	1.6h	16.1h	5.7h	2.8h
	RLSDL	atoms	2,140	805	505	2,730	1,640	770
		time	-	-	45.0h	-	-	28.5h

VI. Conclusion

In this paper, to sparsely represent the temporal-spatial remote sensing big data, we extend the classical K-SVD dictionary learning method. A new object function for big data set is constructed, and the data samples are introduced into the learning process group by group. In the computation, the model mainly focuses on the incremental parts that are hard to sparsely decompose using last dictionary. New atoms are added for current samples data, and an active learning scheme based on a maximum mutual information is employed to determine the initial value of the new atoms. We test the proposed method on two subset data from Landsat satellites: one is a long time sequence on a small area, and the other is a large area over a two-years period. The experiments validated the good performance of the proposed method on both decomposition sparsity and reconstruction precision. We find, while controlling the error of the training process, that the proposed IK-SVD always achieves sparser representation for a temporal-spatial remote sensing big data set. While controlling the sparsity of the training, the precision of the proposed IK-SVD is usually between those of ODL and RLSDL.

References

[1] D. L. Donoho E. J. Cande's, L. Demanet and L. Ying, "Fast discrete curvelet transforms", *Multiscale Modeling and Simulation*, vol. 5, no. 3, pp. 861C899, 2006.

[2] E. LePennec and S. Mallat, "Sparse geometric image representations with bandelets", *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423C438, 2005.

[3] M. Vetterli V. Velisavljevic, B. Beferull-Lozano and P. L. Dragotti, "Directionlets: Anisotropic multidirectional representation with separable filtering", *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1916C1933, 2008.

[4] S. Mallat, "Geometrical grouplets", *Appl. Comput. Harmon. Anal.*, vol. 26, no. 2, pp. 161C180, 2009.

[5] K. Engan, S.O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design", in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, 1999, vol. 5, pp. 2443–2446 vol.5.

[6] Y. Ma R. Vidal and S. Sastry, "Generalized principal component analysis (gpca)", *Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945C1959, 2005.

[7] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation", *Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[8] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images", *Trans. Img. Proc.*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[9] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding", *Journal of Machine Learning Research*, vol. ICML '09, pp. 689–696, 2009.

[10] Karl Skretting and Kjersti Engan, "Recursive least squares dictionary learning algorithm", *Trans. Sig. Proc.*, vol. 58, no. 4, pp. 2121–2130, Apr. 2010.

[11] D. L. Donoho, "Compressed sensing", *IEEE Transanction on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[12] D.L. Donoho, Y. Tsaig, I. Drori, and J-L Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit", *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[13] J. A. Tropp D. Needell, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples", *J. of Appl. and Comput. Harmonic Analysis*, vol. 53, no. 12, pp. 301–321, 2010.

[14] D. Tuia, F. Ratle, F. Pacifici, M. Kanevski, and W. J. Emery, "Active learning methods for remote sensing image classification", *IEEE Trans. Geosci. Remote Sens.*, p. 2218C2232, Jul. 2009.

[15] J. Li, J. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new bayesian approach with active learning", *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3947–3960, Oct. 2011.