Sub clustering K-SVD: Size variable Dictionary learning for Sparse Representations

JianZhou Feng, Li Song, XiaoKang Yang and Wenjun Zhang

Institute of Image Comm. & Information Proc., Shanghai jiaotong University, 200240, Shanghai, China
Shanghai Key Lab of Digital Media Processing and Transmission

ABSTRACT

Sparse signal representation from overcomplete dictionaries have been extensively investigated in recent research, leading to state-of-the-art results in signal, image and video restoration. One of the most important issues is involved in selecting the proper size of dictionary. However, the related guidelines are still not established. In this paper, we tackle this problem by proposing a so-called sub clustering K-SVD algorithm. This approach incorporates the subtractive clustering method into K-SVD to retain the most important atom candidates. At the same time, the redundant atoms are removed to produce a well-trained dictionary. As for a given dataset and approximation error bound, the proposed approach can deduce the optimized size of dictionary, which is greatly compressed as compared with the one needed in the K-SVD algorithm.

Index Terms— Sparse representation, K-SVD, subtractive clustering, OMP

1. INTRODUCTION

Sparse decomposition over a redundant dictionary has been proved as an efficient technique to handle natural images. Suppose a signal $\mathbf{y} \in \mathrm{IR}^n$ has a sparse approximation over a dictionary $\mathbf{D} \in \mathrm{IR}^{nxK}$, which is composed of K atoms $\{\mathbf{d}_j\}_{j=1}^K$, then we can find a linear combination of a "few" atoms from $\mathbf{D}$ that is "close" to the original signal. As for natural image signal $\mathbf{y}$, steerable wavelets, curvelets, contourlets and something like can be the candidates to design dictionaries. In contrast, the learned non-parametric dictionaries are superior to these pre-defined ones for image representation. K-SVD is one of such excellent dictionary learning algorithms, which achieves comparable or even better performance than the state of the art in image denoising, inpainting, compression and so on [1]. However, the total number of atoms (or dictionary size) K-SVD should be known in advance and cannot be selected automatically. Once a relative small dictionary is selected, it might fail to find the sparse linear combination of the given signal. Moreover, a bigger dictionary might introduce redundant atoms, resulting in extra computation burden on sequential processing tasks. Until now, it is still an open issue about how to optimize the dictionary size for sparse representation.

To this point, the most recent work of Mazhar and Gader proposed the EK-SVD algorithm, where the dictionary size can be reduced to a proper value by exploiting the Competitive Agglomeration algorithm to update the dictionary coefficients in K-SVD [2]. Once the proper size has been obtained, EK-SVD can then use the Matching Pursuits algorithm to learn a sparse dictionary accurately. After all, the proper initial size should be assigned to the dictionary. Otherwise, distinct errors or extra computation complexity would be incurred to the learning process of dictionary.

In this paper, we propose a so-called Sub clustering K-SVD algorithm, which characterizes its improvement on K-SVD method in two main aspects: (1) An error-driven mechanism is introduced to the dictionary update stage, achieving a better reconstruction result. (2) Priority of the atoms guides the refinement of the dictionary. Thus the most important atoms are retained and well refined.

The remainder of the paper is organized as follows: Section 2 introduces the related work of Sub clustering K-SVD. Section 3 describes the proposed framework and the implementation details. Experimental results are reported in section 4 and section 5 concludes the paper.

2. PRELIMINARIES

2.1 The K-SVD Algorithm

The K-SVD algorithm can find the dictionary $\mathbf{D}$ that yields sparse representations for a set of training examples. Specifically, this problem can be mathematically described by

$$\min_{\mathbf{D},\mathbf{X}} \{\left\|\mathbf{Y}-\mathbf{DX}\right\|_F^2\} \quad \text{Subject to } \forall i, \| \mathbf{x}_i \|_0 \le T_0 \quad (1)$$

where $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$ is the example set and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is the set of representation coefficients of the signal. $\|A\|_F$ is called the Frobenius norm and is defined as

$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}$, $\|\cdot\|_0$ is the l_0 norm, counts the non-zero entries of a vector (also called *sparsity*).

Like **K**-means, K-SVD use a two phase approach to update **D** and **X** iteratively. In sparse coding stage, **D** is fixed and any pursuit algorithm either Orthogonal Matching Pursuit (OMP) [4] or Basis Pursuit (BP) [5] algorithms can be used to compute $\mathbf{x_i}$ to solve (1). In dictionary update stage, **D** and **X** are assumed to be fixed and only one column $\mathbf{d_k}$ of **D** is updated at a time. Defines the group of examples that use $\mathbf{d_k}$ as:

$$\varpi_k = \{i \mid 1 \le i \le N, x_i(k) \ne 0\} \tag{2}$$

Then compute $\mathbf{E}_k = \mathbf{Y} - \sum_{j \ne k} \mathbf{d}_j \mathbf{x}^j$ and restrict $\mathbf{E}_k$ by choosing the columns corresponding to ϖ_k so that we obtain $\mathbf{E}_k^R$ .Finally, apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\Delta\mathbf{V}^T$ and update $\mathbf{d_k}$ to be first column of **U**, and $\mathbf{x}_k^R$ to be the first column of **V** multiplied by $\Delta(1,1)$.

All dictionary atoms are updated in this way. Iterating through the two steps will produce dictionary that approximates given $\mathbf{y_i}$ sparsely and accurately.

2.2 Subtractive clustering Algorithm

Subtractive clustering (SC) is a simple and effective clustering method to find cluster centers based on a density measure called the mountain function [6]. This technique is similar to mountain clustering, except that instead of calculating the density function at every possible position in the data space, it uses the positions of the data points to calculate the density function, thus reducing the number of calculations significantly. Given the data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathrm{IR}^n$, density measure is defined as

$$D_j = \sum_{i=1}^{N} \exp\left(-\frac{\left\|\mathbf{x}_j - \mathbf{x}_i\right\|_2^2}{r_a^2 / 4} \right) \tag{3}$$

where $\mathbf{r_a}$ is a positive constant representing a neighborhood radius. We call $\mathbf{D_j}$ the potential of $\mathbf{x_j}$. It is easy to see a data point with more points around it will have higher potential.

The first cluster center $\mathbf{x}_{c_1}$ is chosen as the point having the largest potential. Next, the potential of all the data points will be updated as follows

$$D_j = D_j - D_{c_1} \exp\left(-\frac{\left\|\mathbf{x}_j - \mathbf{x}_{c_1}\right\|_2^2}{r_b^2 / 4} \right) \tag{4}$$

where $\mathbf{r_b}$ is a positive constant representing a neighborhood radius that effected badly by the cluster centers. The next cluster center candidate is also selected according to the new potential values. But it is possible to be accepted or rejected as the real center by some rules. If $D_j > \alpha D_{c_1}$, then it is accepted. If $D_j < \beta D_{c_1}$, then it is rejected. α and β are called accept ratio and reject ratio respectively. If $\beta D_{c_1} < D_j < \alpha D_{c_1}$, further calculating $\gamma = \frac{D_j}{D_{c_1}} + \frac{\min_i \left\|\mathbf{x}_j - \mathbf{x}_{c_i}\right\|_2}{r_a}$. If $\gamma > 1$, then the point is accepted, else it is rejected. Every time a new center $\mathbf{c_{new}}$ is found, the potential of all the data points should be updated as in (4) except replacing suffix $\mathbf{c_1}$ by $\mathbf{c_{new}}$.

The merit of this clustering algorithm is it can find clustering centers without predetermined center number. The centers found will have more neighbors than other candidates and be distant enough from each other. In order to use it in following sparse dictionary learning approach, we generalize the subtractive clustering algorithm into a form that every data point has unit norm and given a weight to signify its importance. Thus the initial potential function is converted into

$$D_j = \sum_{i=1}^{n} w_i \exp\left(-\frac{\left\|x_j - x_i\right\|_2^2}{r_a^2 / 4} \right) \tag{5}$$

where w_i is the weight of data x_i and $\|x_i\|_2 = 1$.

3. THE SUB CLUSTERING K-SVD ALGORITHM

The dictionary learning capabilities of K-SVD and the ability to find dominant atoms of subtractive clustering algorithm can be combined to learn better dictionaries with proper size.

In order to show that the centers extracted by subtractive clustering are close to dictionary atoms used to generate the data space, let's consider a simple case. As shown in figure 1, the dictionary **D** contains 3 atoms $\mathbf{d_1}$, $\mathbf{d_2}$ and $\mathbf{d_3}$. The *sparsity* is 2 so that any data points belongs to one of the subspace $\mathbf{S_i}$ (i=1, 2, 3) which are spanned by {$\mathbf{d_2}$, $\mathbf{d_3}$}, {$\mathbf{d_1}$, $\mathbf{d_3}$} and {$\mathbf{d_1}$, $\mathbf{d_2}$} respectively. Point A and point B are two data points of the whole data point set. Point A is a linear combination of $\mathbf{d_2}$ and $\mathbf{d_3}$. Point B is on the direction of $\mathbf{d_3}$. When apply SC to the data points, point B is more likely to be chosen as a center than point A for its higher potential, which is equivalent to having more neighbors. If we define neighbors of data P are data points whose distance from P is less than ***r*** (***r*** is a given radius), B has neighbors from both $\mathbf{S_1}$ and $\mathbf{S_2}$ while all of A's neighbors come from $\mathbf{S_1}$. That means B will approach one of the atoms better than A and should be kept to update later.

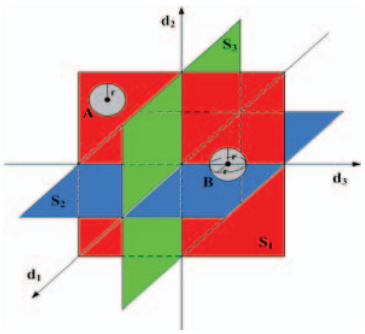


Figure 1 Introductory example

This toy example suggests that SC can be used to improve K-SVD in the following two aspects. Firstly, SC can be used to initialize the dictionary with data points of clustering center. Secondly, we can use SC to pruning similar atoms group or seldom used atoms learned during K-SVD iteration. SC does reduce the dictionary size considerably, but it also has a drawback. When being applied to image patches, it may exclude high frequency atoms compare with those dominant low frequency ones. If we reduce the threshold to retain the high frequency atoms, similar atoms group will also be retained. In order to avoid this situation, it is necessary to category atoms into several groups by their importance and applies SC to each group.

This aim is achieved by extracting groups sequentially in our algorithm. Suppose $\mathbf{D}^{\mathbf{J}}$ is the current learnt dictionary with J atom groups. If the root mean square error (RMSE) of training data by OMP on $\mathbf{D}^{\mathbf{J}}$ is big, which means the number of atoms in $\mathbf{D}^{\mathbf{J}}$ is not big enough to capture the specific structure of training image, we introduce a new group of atoms $\mathbf{G}^{\mathbf{J+1}}$ to approximate training data. Here we initialize $\mathbf{G}^{\mathbf{J+1}}$ by applying SC to the residuals of half-*sparsity* ($\boldsymbol{T}_0/2$) approximations. $\boldsymbol{T}_0/2$ is chosen as a trade-off since keeping distance from $\boldsymbol{T}_0$ can make the impact of atoms that provide trivial contribution to $\hat{y}_i$ on $\hat{r}_i$ reduced while small value makes most atoms in $\mathbf{G}^{\mathbf{J+1}}$ close to atoms in $\mathbf{D}^{\mathbf{J}}$. After that, the new dictionary $\mathbf{D}^{\mathbf{J+1}}$ is improved by K-SVD and atoms in $\mathbf{G}^{\mathbf{J+1}}$ is updated by SC to prune possible similar atoms wherein weight values are by product of K-SVD. Multiple iterations make the dictionary learn more and more precise structure from the training image until the expected RMSE reached.

It is noted that our sub clustering K-SVD algorithm is different from EK-SVD in two aspects: (1) EK-SVD learned dictionaries are the same as K-SVD when the dictionary size is set correctly. Our algorithm learns more accurate and auto sorted dictionaries by using SC to extract atom candidates. (2) The dictionary size of EK-SVD decreases all the time. For our algorithm, it increases most of the time with improving approximation quality and decrease only to prune possible similar atoms.

A full description of the algorithm is given below:

Inputs: data points Y= $\{y_i\}_{i=1}^{N}$, expected RMSE E.

Initializing:

For i=1,…,N, $w_i = \|y_i\|_2$ and $d_i^0 = \frac{y_i}{w_i}$

Initial dictionary and first group of atoms $D^0=G^0=SC(\{w_i\},\{d_i^0\})$

Set initial OMP RMSE=E^0. Set J=0, Iter=$Iter_{max}$.

Repeat until Iter=0

- $(X_J,count,E^{real})=OMP(Y,D^J,E^J,T_0)$
  count: the number of image patches whose sparsity > T_0 under noise level E^J
  E^{real} : the real RMSE after OMP
- If (count > N/10 and E^{real}> E)
  time to find new group of atoms
  Residuals $\hat{r}_i = y_i - \hat{y}_i$, $\hat{y}_i$ is the approximation of y_i using $T_0/2$ atoms
  For i=1,…,N, $w_i = \|\hat{r}_i\|_2$ and $d_i^{J+1} = \frac{\hat{r}_i}{w_i}$
  $G^{J+1} = SC(\{w_i\},\{d_i^{J+1}\})$
  $G^{J+1} = G^{J+1} \setminus D^J$: an atom in G^{J+1} too close to an atom in D^J should be cut
  $D^{J+1} = D^J \cup G^{J+1}$ and $index = \{i \mid d_i \in G^{J+1}\}$
  $X_{J+1}=OMP(Y,D^{J+1},E^J,T_0)$
  Else
  $D^{J+1}=D^J$, $G^{J+1}=G^J$ and $X_{J+1}=X_J$
- If count < N/10, $E^{J+1}=\max(\delta E^J, E)$, $\delta<1$ is the learning step
- If $E^{real}<E$, Iter = Iter-1
- $(X_{J+1}, D'^{J+1}) = KSVD(X_{J+1}, D^{J+1})$
- $G'^{J+1} = \{d'_i \mid i \in index\}$
- For $i \in index$ $w_i = \|x^i_{J+1}\|_1$ and $G'^{J+1} = SC(\{w_i\}, G'^{J+1})$
- $D^{J+1} = \{d'_i \mid i \notin index\} \cup G'^{J+1}$

Set J=J+1

The sub clustering K-SVD algorithm

4. EXPERIMENTAL RESULTS

Simulation results: we conduct the synthetic tests same as in [1] and compare the result with K-SVD.

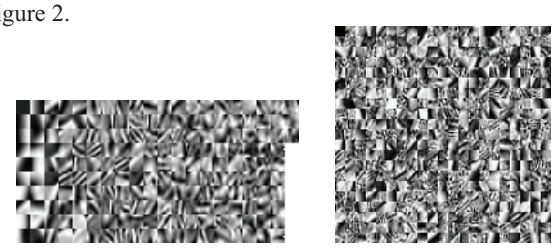
We randomly choose a dictionary $D \in R^{20\times 50}$ and form a data set Y= $\{y_i\}_{i=1}^{1500}$ whose atoms are linear combination of 3 dictionary atoms from $\mathbf{D}$. White Gaussian noise with varying signal-to-noise ratio(SNR) was added to the resulting data points. For each noise level, 50 trails were conducted. In these tests, the parameters of sub clustering K-SVD are set as follows: r_a=0.5, r_b=1, α=0.5, β=0.15, E^0=0.1, δ =0.9, $\boldsymbol{T}_0$=10, $Iter_{max}$=20 and RMSE E set according to the noise level. For K-SVD, we set the dictionary size K = 50, *sparsity* $\boldsymbol{T}_0$ = 3, iteration number $Iter_{max}$=20.

Results show that the estimated sizes are between 50 ± 1 and 92.5% of them are equal to 50. The rate of detected atoms is more than 95% and is 14%, 13%, 13%, and 24% higher than K-SVD for SNR levels of no noise, 30db, 20db, and 10db case. It means sub clustering K-SVD

converges to the real dictionary atoms correctly and within fewer iteration times.

Natural image experiments: The following experiments show that sub clustering K-SVD can learn proper dictionaries and achieve slightly lower error than K-SVD.

Experiment 1: The training data consisted of 10000 block patches of size 8*8 pixels randomly chosen from a training image. The testing data is obtained by dividing the same image into successive 8*8 blocks. For sub clustering K-SVD, the RMSE is set to 0.018. The other parameters are set the same as in the synthetic experiment. The K-SVD parameters are set the same as in [1]: K=441, $\boldsymbol{T}_0$=10 and Iter_{max}=15. The dictionary learned from *lena* is shown in figure 2.



(a) Sub clustering K-SVD **(b) K-SVD**

Figure 2 dictionaries learned from *lena* training data

Table 1: Dictionaries comparison

	Lena		*Baboon*	
Algorithm	size	RMSE	size	RMSE
Sub clustering K-SVD	193	0.0124	2779	0.0178
K-SVD	441	0.0114	441	0.0368

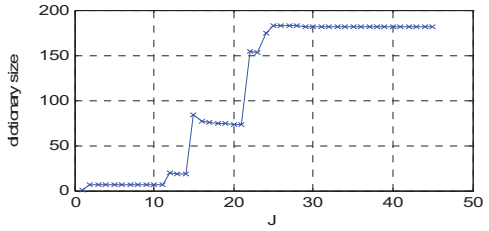
As shown in table 1, the RMSE is close with the dictionary size reduce to 44% of K-SVD size for *lena*. For *Baboon*, our method obtains 2779 atoms to approach target RMSE while the result of K-SVD with 441 atoms is unbearable. The different size learnt shows our method can adaptively find a proper size dictionary. However, it is hard for K-SVD and EK-SVD to set an initial dictionary size.

Experiment 2: we use a publically available face images database [7] as our image source. In the database, there are 15 individuals and 11 images per subject. We randomly extract 10000 block patches of size 8*8 from images of the first 5 subjects as training data and another 20000 block patches from the last 10 subjects as testing data. The parameters used in the experiments are the same as in experiments 1 except that set $\boldsymbol{T}_0$ = 6 as in [2]. The iteration number used in K-SVD is set to 20 and the dictionary size is set to 441 and 179 as in [2]. The result is listed in table 2.

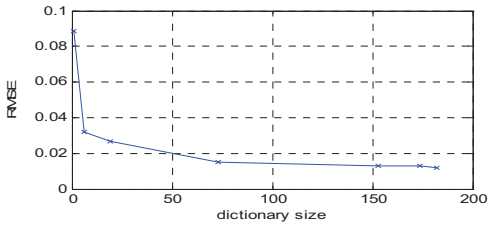
Table 2: Performance comparison

Algorithm	Sub clustering K-SVD	K-SVD	K-SVD
Size	182	441	179
RMSE	0.0115	0.0115	0.0131

Table 2 shows that our method find dictionary size 182 similar to EK-SVD's 179 but with smaller RMSE. In Figure 3, the dictionary size increase stepwise as new atom groups introduced sequentially and the last 20 iterations are just for finding more accurate atoms.



(a) Dictionary size *vs* iteration



(b) RMSE *vs* dictionary size

Figure 3 Sub Clustering K-SVD Iterations

5. CONCLUSIONS

In this paper, we present a so-called sub clustering K-SVD algorithm to provide a useful tool for estimating the proper size of dictionary in sparse signal representation. As compared with K-SVD method, a rather smaller dictionary is needed to satisfy the given error bound.

6. ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (60702044, 60625103 and 60632040).

7. REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", *IEEE Trans. Signal Processing*, 54(11), pp.4311-4322 November 2006.

[2] Raaziz Mazhar, Paul D.Gader, "EK-SVD: Optimized Dictionary Design for Sparse Representations", *In Proceedings of 19th International Conference on Pattern Recognition*, 2008.

[3] Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, pp. 209-219,1994.

[4] J.A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):2231–2242, October 2004.

[5] S.S.Chen,D.L.Donoho andM.A.Saunders "Atomic decomposition by basis pursuit." *SIAM Review*, 43(1):129–159, 2001.

[6] Ronald R.Yager and Dimitar P. Filev, "Approximate Clustering Via the Mountain Method", *IEEE Trans. on Syst. Man and Cyb.*, vol. 24, No. 8, August 1994.

[7] Yale Face Database. http://cvc.yale.edu/projects/yalefaces/yalefaces.html