Dictionary training for sparse representation as generalization of K-means clustering

Sahoo, Sujit Kumar; Makur, Anamitra

2013

Sahoo, S. K., & Makur, A. (2013). Dictionary Training for Sparse Representation as Generalization of K-Means Clustering. IEEE Signal Processing Letters, 20(6), 587-590.

https://hdl.handle.net/10356/96655

https://doi.org/10.1109/LSP.2013.2258912

Dictionary Training for Sparse Representation as Generalization of K-means Clustering

Sujit Kumar Sahoo, *Member, IEEE* and Anamitra Makur, *Senior Member, IEEE*

***Abstract*—Recent dictionary training algorithms for sparse representation like K-SVD, MOD, and their variation are reminiscent of K-means clustering, and this letter investigates such algorithms from that viewpoint. It shows: though K-SVD is sequential like K-means, it fails to simplify to K-means by destroying the structure in the sparse coefficients. In contrast, MOD can be viewed as a parallel generalization of K-means, which simplifies to K-means without perturbing the sparse coefficients. Keeping memory usage in mind, we propose an alternative to MOD; a sequential generalization of K-means (SGK). While experiments suggest a comparable training performances across the algorithms, complexity analysis shows MOD and SGK to be faster under a dimensionality condition.**

***Index Terms*—dictionary training, K-means, K-SVD, MOD.**

I. Introduction

In recent years sparse representation has emerged as a new tool for signal processing. Given a dictionary $\mathbf{D} \in \mathbb{R}^{n\times K}$ containing prototype signal-atoms $\mathbf{d}_k \in \mathbb{R}^n$ for $k = 1, \ldots, K$, the goal of sparse representation is to represent a signal $\mathbf{y} \in \mathbb{R}^n$ as a linear combination of a small number of atoms $\hat{\mathbf{y}} = \mathbf{D}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^K$ is the sparse representation vector and $\|\mathbf{x}\|_0 = m \ll n$ ($\|.\|_0$ is ℓ_0 norm). Dictionaries that better fit such a sparsity model, can either be chosen from a prespecified set of linear transforms (e.g. Fourier, Cosine, Wavelet, etc.) or can be trained on a set of training signals.

Given a set of signals, a trained $\mathbf{D}$ will always produce a better sparse representation in comparison to traditional parametric bases. This is because, for a set of training signals $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$, $\mathbf{D}$ is trained to minimize the representation error, $\mathbf{E} = \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ are the sparse representations and $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} {A_{ij}}^2}$ is the Frobenius norm of a matrix $\mathbf{A}$. Noting that the error minimization depends both on $\mathbf{X}$ and $\mathbf{D}$, the solution is obtained iteratively by alternating between *sparse coding* (for $\mathbf{X}$) and *dictionary update* (for $\mathbf{D}$) as the following.
1) *Sparse coding stage*: Obtain $\mathbf{X}^{(t)}$ for each $\mathbf{y}_i$ in $\mathbf{Y}$ as

$$\forall_i \mathbf{x}_i = \arg\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}^{(t)}\mathbf{x}_i\|_2^2 : \|\mathbf{x}_i\|_0 \leq m_{\max} \qquad (1)$$

where $m_{\max}$ is the admissible number of coefficients.
2) *Dictionary update stage*: For the obtained $\mathbf{X}^{(t)}$, update $\mathbf{D}^{(t)}$ such that

$$\mathbf{D}^{(t+1)} = \arg\min_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{(t)}\|_F^2. \qquad (2)$$

This work is funded by the AcRF project RG 27/10. We thank the reviewers for their useful feedbacks. We also acknowledge M. Aharon and M. Elad for the reproducibility of K-SVD results.

At present, a sequential dictionary training algorithm called K-SVD has become a benchmark in dictionary training [1]. This algorithm derives its name from its sequential update of K atoms using singular value decomposition (SVD), and it is claimed that K-SVD is advantageous over MOD in terms of speed and accuracy. The Method of Optimal Direction (MOD) for frame design is an earlier attempt in the direction of dictionary training, which updates all the atoms in parallel as the minimum mean square error (MMSE) solution for given $\mathbf{Y}$ and $\mathbf{X}^{(t)}$ [2].

Such iterative algorithms as above are reminiscent of long-known K-means clustering used for codebook design (dictionary training) in vector quantization (VQ) [3]. It is an extreme form of sparse representation, where the dictionary $\mathbf{D}$ is termed as codebook, and the coefficient vector is restricted/constrained to the trivial basis in $\mathbb{R}^K$, that is, $\mathbf{x} = e_k$ has all 0s except 1 in the k^{th} position. To minimize the representation error, a VQ codebook is typically trained using K-means clustering algorithm. It is an iterative process similar to dictionary training which alternates between finding $\mathbf{X}$ and updating $\mathbf{D}$.
1) *Sparse coding (encoding) stage*: This stage involves finding the index $k = \arg\min_j \|\mathbf{y}_i - \mathbf{D}^{(t)}e_j\|_2^2$, so that the sparse representation for $\mathbf{y}_i$ becomes $\mathbf{x}_i = e_k$. Likewise, the representation $\mathbf{X}^{(t)}$ for each training signal in $\mathbf{Y}$ is obtained. As a result, $\mathbf{Y}$ is partitioned into K clusters, $\{1 : N\} = \{R_1^{(t)} \cup R_2^{(t)} \cdots \cup R_K^{(t)}\}$, where each cluster $R_k^{(t)} = \{i : \hat{\mathbf{y}}_i = \mathbf{D}^{(t)}e_k\}$.
2) *Dictionary update (codebook design) stage*: Due to disjoint clustering, the global problem in (2) becomes local minimization for each individual signal-atom (codeword), $\mathbf{d}_k^{(t+1)} = \arg\min_{\mathbf{d}_k} \sum_{i\in R_k^{(t)}} \|\mathbf{y}_i - \mathbf{d}_k\|_2^2 = \frac{1}{|R_k^{(t)}|} \sum_{i\in R_k^{(t)}} \mathbf{y}_i$. As a result, it allows independent update of K atoms sequentially.

In this letter we investigate how K-means clustering may be generalized to sparse representation. In the next sections, we elaborate on K-SVD and MOD, and discuss their analogy to K-means. It is shown that K-SVD in its present form fails to retain any structured sparsity such as VQ, and as a result does not simplify to K-means. Use of SVD interferes with the sparse coding, and also restricts the signal-atoms to unit norm. In contrast, it is shown that MOD retains any structured sparsity such as VQ, and simplifies to K-means, hence it may be claimed as a parallel generalization of K-means clustering. However, in many practical scenarios sequential algorithms are desirable to operate with minimum computational resources. Thus a sequential alternative to MOD is proposed, which is referred as SGK. In the subsequent sections the computational complexity is analyzed, and the training performances are

examined experimentally. They suggest, a very much comparable training performance across the algorithms, and under a dimensionality condition MOD takes the least computation time followed by SGK.

II. K-SVD

In the dictionary update stage, K-SVD breaks the global minimization problem (2) into K sequential minimization problems [1]. It considers each column $\mathbf{d}_k$ in $\mathbf{D}$ and its corresponding row of coefficients $X_{\text{row}k}$ in $\mathbf{X}$. Thus the error term in (2) may be written as $\|\mathbf{E}^{(t)}\|_F^2 = \|(\mathbf{Y} - \sum_{j\neq k} \mathbf{d}_j^{(t)} X_{\text{row}j}^{(t)}) - \mathbf{d}_k^{(t)} X_{\text{row}k}^{(t)}\|_F^2$. The quest is for the $\mathbf{d}_k X_{\text{row}k}$ which is closest to $\mathbf{E}_k^{(t)} = \mathbf{Y} - \sum_{j\neq k} \mathbf{d}_j^{(t)} X_{\text{row}j}^{(t)}$,

$$\{\mathbf{d}_k^{(t+1)}, \hat{X}_{\text{row}k}^{(t)}\} = \arg \min_{\mathbf{d}_k, X_{\text{row}k}} \|\mathbf{E}_k^{(t)} - \mathbf{d}_k X_{\text{row}k}\|_F^2. \quad (3)$$

In [1] SVD is used to find the closest rank-1 matrix (in Frobenius norm) that approximates $\mathbf{E}_k^{(t)}$ subject to $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$. SVD decomposition is done on $\mathbf{E}_k^{(t)} = \mathbf{U}\Delta\mathbf{V}^T$. $\mathbf{d}_k^{(t+1)}$ is taken as the first column of $\mathbf{U}$, and $\hat{X}_{\text{row}k}^{(t)}$ is taken as the first column of $\mathbf{V}$ multiplied by the first diagonal element of Δ.

Note that different from (2), both $\mathbf{d}_k$ and $X_{\text{row}k}$ are updated in K-SVD dictionary update stages (apart from updating $X_{\text{row}k}$ in the sparse coding stage). Unlike K-means, if each signal-atom is updated alone, the resulting $\mathbf{D}^{(t+1)}$ may diverge. This is because there exists a considerable amount of overlap among $X_{\text{row}k}^{(t)}$'s (clusters R_k), so modifying an atom affects other atoms. In order to take care of this overlap, before updating the next atom, both $\{\mathbf{d}_k^{(t)}, X_{\text{row}K}^{(t)}\}$ are replaced with $\{\mathbf{d}_k^{(t+1)}, \hat{X}_{\text{row}K}^{(t)}\}$. The process is repeated for all K atoms. We should note that K-SVD is an interdependent sequential update procedure.

However, there are few matters of concern over the simultaneous update of $\{d_k, X_{\text{row}k}\}$ in (3) using SVD.

1) *Loss of sparsity*: As there is no sparsity control term $\|X_{\text{row}k}^{(t)}\|_0$ in SVD, the least square solution $\hat{X}_{\text{row}k}^{(t)}$ may contain all nonzero entries, which will result in a nonsparse $\hat{\mathbf{X}}^{(t)}$.

2) *Loss of structure*: Similarly, if any *structured/constrained sparsity* is used in the sparse coding stage of the dictionary training, this structure may also not be retained in both the solutions obtained.

3) *Normalized dictionary*: The use of SVD limits the usability of this dictionary training algorithm only to the settings of unit norm atoms, $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$.

To address the *Loss of sparsity* issue, K-SVD restricts the minimization problem of (3) to only the set of training signals $\mathbf{Y}_k^{(t)} = \{\mathbf{y}_i : X_{\text{row}k}^{(t)}(i) \neq 0\}$. Defining an index set $R_k^{(t)} = \{i : 1 \leq i \leq N, X_{\text{row}k}^{(t)}(i) \neq 0\}$, SVD decomposition is done on only a part of $\mathbf{E}_k^{(t)}$ that keeps the columns from this index set. However, the *Loss of structure* issue still remains unaddressed. Let's take an example of a sparse coder with additional structure/constraint $S(\mathbf{x}_i)$,

$$\mathbf{x}_i = \arg \min_{\mathbf{x}_i} \{\|\mathbf{y}_i - \mathbf{D}^{(t)}\mathbf{x}_i\|_2^2 + S(\mathbf{x}_i)\} : \|\mathbf{x}_i\|_0 \leq m_{\max} \quad (4)$$

K-SVD in its present form updates both $\{\mathbf{d}_k, X_{\text{row}k}\}$ using SVD, which cannot take care of the additional structure/constraint $S(X_{\text{row}k})$. Similarly, it fails to simplify to K-means for the VQ as elaborated in the next paragraph. Alongside, the issue of *Normalized dictionary* brings further complication to the usability of K-SVD in VQ.

In order to verify K-SVD as a generalization of K-means clustering, use K-SVD to update the codebook for VQ, where $\{\mathbf{d}_k^{(t+1)}, \hat{X}_{\text{row}k}^{(t)}\}$ is obtained using SVD decomposition. First thing to note that, use of SVD will result in $\|\mathbf{d}_k^{(t+1)}\|_2 = 1$ which is not same as the K-means. Secondly, VQ is a binary *structured/constrained sparsity* with only 0 and 1 entries. Thus the SVD decomposition is done on the part of $\mathbf{E}_k^{(t)}$ with column indices $R_k^{(t)} = \{i : 1 \leq i \leq N, X_{\text{row}k}^{(t)}(i) = 1\}$. Even if $\hat{X}_{\text{row}k}^{(t)}$ on the restricted index set $R_k^{(t)}$ is formed by an oracle scaling of obtained $\mathbf{V}$ from SVD, all of its entries cannot be guaranteed to be 1. This is a classical example of discussed *Loss of structure* issue of K-SVD, which destroys the binary structure imposed by VQ. Thus, it can be concluded that K-SVD as presented in [1] is not a generalization of K-means.

III. MOD

In the dictionary update stage, MOD analytically solves the minimization problem (2) [2]. The quest is for a $\mathbf{D}$ that minimizes the error $\|\mathbf{E}^{(t)}\|_F^2 = \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{(t)}\|_F^2$ for the obtained $\mathbf{X}^{(t)}$. Thus taking the derivative of $\|\mathbf{E}^{(t)}\|_F^2$ with respect to $\mathbf{D}$, and equating with 0 gives the relationship: $\frac{\partial}{\partial \mathbf{D}} \|\mathbf{E}^{(t)}\|_F^2 = -2(\mathbf{Y} - \mathbf{D}\mathbf{X}^{(t)}){\mathbf{X}^{(t)}}^T = 0$, leading to [1]

$$\mathbf{D}^{(t+1)} = \mathbf{Y}{\mathbf{X}^{(t)}}^T \left(\mathbf{X}^{(t)}{\mathbf{X}^{(t)}}^T\right)^{-1}. \quad (5)$$

In each iteration, MOD obtains $\mathbf{X}^{(t)}$ for a given $\mathbf{D}^{(t)}$, and updates $\mathbf{D}^{(t+1)}$ using (5). MOD doesn't require the atoms of the dictionary to be unit norm. However, if it is required by the sparse coder, the atoms of $\mathbf{D}^{(t+1)}$ may be normalized to unit norm.

It is interesting to note that MOD is a coder independent dictionary training algorithm, which can be used for all sparse representation applications. Let's take an example of sparse coder with additional structure/constraint $S(\mathbf{x}_i)$ as in (4). As MOD updates $\mathbf{D}$ independent of $\mathbf{X}$, the presence of $S(\mathbf{X})$ will not affect the minimization in (5). Thus, codebook update for VQ using MOD simplifies to K-means as elaborated in the next paragraph.

In the case of VQ, $X_{\text{row}k}$ has all 0 entries except 1s at the positions $i \in R_k$, that is, when $\hat{\mathbf{y}}_i = \mathbf{D}e_k$. As it produces disjoint clusters, rows of $\mathbf{X}$ will be orthogonal to each other ($\forall_{j\neq k} X_{\text{row}j}{X_{\text{row}k}}^T = 0$). This gives us $\mathbf{X}\mathbf{X}^T = \text{diag}\{|R_1|, \ldots, |R_K|\}$, where $|R_k|$ is the number of training signals associated with signal-atom $\mathbf{d}_k$. Similarly, it can be written that $\mathbf{Y}\mathbf{X}^T = \left[\sum_{i\in R_1} \mathbf{y}_i, \ldots, \sum_{i\in R_K} \mathbf{y}_i\right]$, because $\mathbf{Y}X_{\text{row}k}^T = \sum_{i\in R_k} \mathbf{y}_i$. Thus the dictionary update of MOD as in (5) simplifies to the dictionary update of K-means clustering.

In other words, minimization of the representation error of K-means clustering generalizes to MOD when the trivial basis of VQ is extended to arbitrary sparse representation with an admissible number of coefficients $m_{\max}$. However, it is a parallel update algorithm in contrast to K-means, which may require more resources (e.g. memory, cache and higher bit processors) to execute for large K and N.

TABLE I
COMPARISON OF EXECUTION TIME (IN MILLISECOND)

m	$\mathcal{T}_{K\text{-SVD}}$	$\mathcal{T}_{K\text{-SVDa}}$	$\mathcal{T}_{\text{MOD}}$	$\mathcal{T}_{\text{SGK}}$
3	139.2	12.0	0.52	4.8
4	145.6	13.4	0.61	5.7
5	151.6	15.0	0.71	6.9

IV. A Sequential Generalization of K-means

Though MOD is suitable for all kind of spare representation applications, irrespective of constraints on sparse coefficient and dictionary, it may demand more computational resource to operate. In contrary, sequential algorithms like K-SVD and K-means can manage with lesser resources. This leads naturally to the possibility to generalize K-means sequentially for general purpose sparse representation application. Thus, we propose a modification to the problem formulation in (3). If we keep $X_{\text{row}k}^{(t)}$ unchanged, both concerns of loss of sparsity and loss of structure of $\hat{\mathbf{X}}^{(t)}$ will no longer be there. Thus we pose the sequential update problem as

$$\mathbf{d}_k^{(t+1)} = \arg\min_{\mathbf{d}_k} \|\mathbf{E}_k^{(t)} - \mathbf{d}_k X_{\text{row}k}^{(t)}\|_F^2. \quad (6)$$

The solution to (6) can be obtained in the same manner as (5)

$$\mathbf{d}_k^{(t+1)} = \mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T \left(X_{\text{row}k}^{(t)} {X_{\text{row}k}^{(t)}}^T\right)^{-1}. \quad (7)$$

The overlap among $X_{\text{row}k}^{(t)}$'s (clusters R_k) is taken care of by replacing $\mathbf{d}_k^{(t)}$ with $\mathbf{d}_k^{(t+1)}$ before updating the next atom in the sequence. This process is repeated for all K atoms sequentially similar to K-means. Recall that it is called SGK.

Similar to MOD, SGK does not constrain the signal-atoms to be unit norm. If required by the sparse coder, all the atoms can be normalized after updating the entire dictionary. Like MOD, the update equation of SGK (7) is independent of the sparse coder, which remains unaffected by the presence of any additional structure/constraint $S(X_{\text{row}k})$ as per the exemplar coder (4). Thus, codebook update for VQ using SGK simplifies to K-means as follows.

In the case of VQ, it can be shown that $\mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T = \mathbf{Y} {X_{\text{row}k}^{(t)}}^T - \sum_{j\neq k} \mathbf{d}_j^{(t)} X_{\text{row}j}^{(t)} {X_{\text{row}k}^{(t)}}^T = \sum_{i\in R_k^{(t)}} \mathbf{y}_i$ because $\mathbf{Y} {X_{\text{row}k}^{(t)}}^T = \sum_{i\in R_k} \mathbf{y}_i$ and $\forall_{j\neq k} {X_{\text{row}j}}^T X_{\text{row}k} = 0$. Thus (7) gives $\mathbf{d}_k^{(t+1)} = \frac{1}{|R_k^{(t)}|}\sum_{i\in R_k^{(t)}} \mathbf{y}_i$, which is same as K-means.

V. Complexity Analysis

We are interested in the complexity analysis of the dictionary update stage alone. In order to compute the complexity, let's assume that each training signal of length n has a sparse representation with m nonzero entries, and $\mathbf{Y}$ contains N such training signals.

In the process of updating $\mathbf{d}_k$ using K-SVD, we need $2n(m-1)|R_k^{(t)}|$ floating point operations (flop) to compute $\mathbf{E}_k^{(t)} = \mathbf{Y} - \sum_{j\neq k} \mathbf{d}_j X_{\text{row}j}^{(t)}$ in the restricted index set $R_k^{(t)}$, because the columns of the sparse representation matrix $\{\mathbf{x}_i : i \in R_k\}$ have only $(m-1)$ nonzero entries to be multiplied with $d_{j\neq k}$. Then performing SVD on $n \times |R_k^{(t)}|$ matrix $\mathbf{E}_k^{(t)}$ requires $2|R_k^{(t)}|n^2 + 11n^3$ flops [4], and $|R_k^{(t)}|$ flops to compute $\hat{X}_{\text{row}k}^{(t)}$ by multiplying the first column of $\mathbf{V}$ with the first diagonal element of Δ. This gives a total of $2n(m-1)|R_k^{(t)}| + 2n^2|R_k^{(t)}| + 11n^3 + |R_k^{(t)}|$ flops to update one atom in $\mathbf{D}^{(t)}$. Thus the flops needed for K-SVD will be the sum over all K atoms,

$$\mathcal{T}_{K\text{-SVD}} = 2nm^2N + 2mn^2N + 11n^3K + mN - 2nmN \quad (8)$$

because $\mathbf{X}^{(t)}$ contains $\sum_k |R_k^{(t)}| = Nm$ nonzero elements.

Though SVD gives the closest rank-1 approximation, this step makes K-SVD very slow. Thus in [5] an approximation algorithm is proposed to replace the SVD step, which makes it faster. In approximate K-SVD, (3) is approximately determined in two steps: 1) $\mathbf{d}_k^{(t+1)} = \mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T / \|\mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T\|_2$; 2) $\hat{X}_{\text{row}k}^{(t)} = {\mathbf{d}_k^{(t+1)}}^T \mathbf{E}_k^{(t)}$. Thus we need $n(2|R_k^{(t)}|-1)$ operations to compute $\mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T$, approximately $3n$ operations to normalize the atom, and $|R_k^{(t)}|(2n-1)$ operations to compute ${\mathbf{E}_k^{(t)}}^T \mathbf{d}_k^{(t+1)}$. Including $2n(m-1)|R_k^{(t)}|$ operations to compute $\mathbf{E}_k^{(t)}$, it needs a total of $2n(m+1)|R_k^{(t)}| + 2n - |R_k^{(t)}|$ flops to update one atom in $\mathbf{D}^{(t)}$. Thus the flops needed for approximate K-SVD will be the sum over all K atoms,

$$\mathcal{T}_{K\text{-SVDa}} = 2nm^2N + 2nmN + 2nK - mN \quad (9)$$

The number of operations required for solving (5) can be derived in the case of MOD. It is known that $\mathbf{X}^{(t)}$ is sparse and contains only $N.m$ nonzero entries. Thus, the cumulative number of operations required to perform the multiplication $\mathbf{Y}{\mathbf{X}^{(t)}}^T$ will sum up to $2nmN - nK$. Likewise, $\mathbf{X}^{(t)}{\mathbf{X}^{(t)}}^T$ will need $2m^2N - K^2$ operations. $\mathbf{X}^{(t)}{\mathbf{X}^{(t)}}^T$ is a symmetric positive definite matrix[1], thus Cholesky factorization can be used to solve the linear inverse problem (5). Cholesky factorization factors $A \in \mathbb{R}^{K\times K}$ as $A = LL^T$ in $\frac{K^3}{3}$ operations, and for solving the linear inverse problem for n vectors it needs $2nK^2$ operations, which sum up to $2nK^2 + \frac{1}{3}K^3$ operations [4]. Thus the total flop count for MOD will be

$$\mathcal{T}_{\text{MOD}} = 2nmN + 2m^2N + 2nK^2 + \frac{K^3}{3} - nK - K^2. \quad (10)$$

Similarly, for SGK we need $2n(m-1)|R_k^{(t)}|$ operations to compute $\mathbf{E}_k^{(t)}$, $n(2|R_k^{(t)}|-1)$ operations are needed to compute $\mathbf{E}_k^{(t)} {X_{\text{row}k}^{(t)}}^T$, approximately $2|R_k^{(t)}|-1$ operations are needed to compute $X_{\text{row}k}^{(t)} {X_{\text{row}k}^{(t)}}^T$, and n operations are needed for the division. This gives a total of $2nm|R_k^{(t)}| + 2|R_k^{(t)}| - 1$ operations needed to update one atom in $\mathbf{D}^{(t)}$. Thus the total flops required for SGK will be the sum over all K atoms,

$$\mathcal{T}_{\text{SGK}} = 2nm^2N + 2mN - K. \quad (11)$$

The complexity expressions give a sense that MOD is the least complex, which contains only 3rd order terms. However for a fair comparison, let's express all the variables in terms of K. In general, the signal dimension $n = O(K)$, and the number of training samples $N = O(K^{1+a})$, where $a \geq 0$. Therefore, a condition for minimum complexity may

[1] $\mathbf{X}^{(t)}{\mathbf{X}^{(t)}}^T$ can be positive semi definite if any atom from $\mathbf{D}^{(t)}$ is completely unused. In that case we can remove those atoms from $\mathbf{D}^{(t)}$ and the corresponding row from the sparse representation matrix.

TABLE II
AVERAGE NO. OF ATOMS RETRIEVED BY DICTIONARY TRAINING

	10 dB	20 dB	30 dB	No Noise	
K-SVD	37.14	46.14	46.74	47.10	$m = 3$
K-SVDa	37.76	45.68	46.04	46.42	
MOD	36.54	45.50	46.86	46.48	
SGK	36.78	46.16	46.46	46.48	
K-SVD	18.10	46.50	47.60	47.36	$m = 4$
K-SVDa	16.92	45.48	46.60	46.56	
MOD	17.90	45.70	46.54	46.78	
SGK	17.80	46.06	46.94	46.78	
K-SVD	00.94	45.40	46.38	46.64	$m = 5$
K-SVDa	00.76	45.12	46.96	46.94	
MOD	01.22	45.88	46.26	47.12	
SGK	00.82	45.82	46.68	46.80	

be derived by taking sparsity $m = O(K^b)$. It can be found that $\min \mathcal{T}_{K\text{-SVD}} = O(K^4)$, and $\min \mathcal{T}_{\text{MOD}} = O(K^3)$, whereas $\forall_{b\geq 0}\ \mathcal{T}_{K\text{-SVDa}} = \mathcal{T}_{\text{SGK}} = O(K^{2+2b+a})$. Thus MOD remains least complex as long as $b \geq 0.5(1-a)$, and this dimensionality condition is very likely in practical situations. Therefore it can safely be stated, $\mathcal{T}_{\text{MOD}} \leq \mathcal{T}_{\text{SGK}} < \mathcal{T}_{K\text{-SVDa}} \ll \mathcal{T}_{K\text{-SVD}}$. Alongside, the execution time of all algorithms in Matlab environment[2] is compared in Table I, for $n = 20, K = 50, N = 1500$, and various m, which agrees with the above analysis. It also reflects that being a parallel update procedure, MOD's execution time reduces by a factor of $O(K)$.

VI. Synthetic Experiment

Similar to [1], we apply K-SVD, approximate K-SVD, MOD and the sequential generalization on synthetic signals to test how well they recover the original dictionary that generated the signal. A random matrix $\mathbf{D}$ (later referred as generating dictionary) of size 20×50 is generated with i.i.d. uniformly distributed entries. As K-SVD can only operate on a normalized dictionary, each column is normalized to unit ℓ_2 norm. Then 1500 training signals $\{\mathbf{y}_i\}_{i=1}^{1500}$ of dimension 20 are produced, each created by a linear combination of m atoms at random locations with i.i.d. coefficients. White Gaussian noise is added to the resulting signals so that each training signal has the same signal to noise ratio (SNR).

In all the algorithms, the dictionaries are initialized with the same set of training signals. As per the suitability of K-SVD, an unconstrained sparse coding is done using orthogonal matching pursuit (OMP), which produces best m-term approximation for each signal [6]. All dictionary training algorithms are iterated $9m^2$ times for sparsity level m.

The trained dictionaries are compared against the known generating dictionary in the same way as in [1]. The mean number of atoms retrieved over 50 trials are computed for each algorithm at different sparsity levels $m = 3, 4, 5$ with additive noise SNR $= 10, 20, 30, \infty$ dB. The results are tabulated in Table II, which shows marginal difference among all the algorithms. In order to show convergence of the algorithms, the average number of atoms retrieved after each iteration is shown in Fig. 1 for one of the SNR from Table II. Given their comparable performance but differing complexity, it may be concluded that MOD is the better choice for dictionary training unless sequential update becomes essential, in which case the sequential generalization should be chosen. Both MOD and SGK can be used in all sparse representation applications irrespective of constraints on dictionary and sparse coder.

[2]Matlab was running on a 64 bit OS with 8GB memory and 3.1GHz CPU.

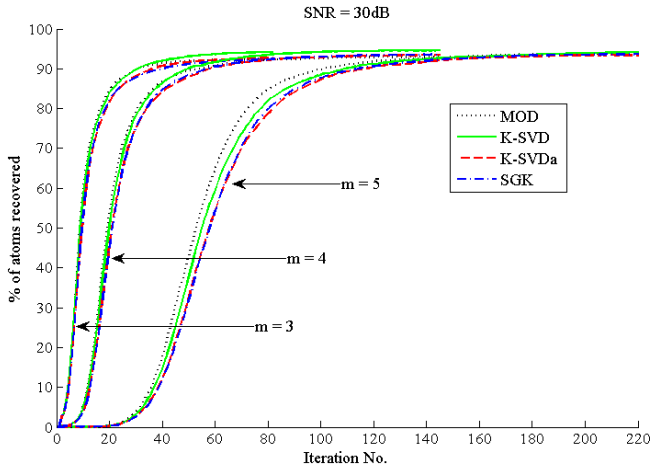

Fig. 1. Average number of atoms retrieved after each iteration for different values of m at SNR $= 30$ dB

VII. Discussion

Existing dictionary training algorithms MOD, K-SVD, and approximate K-SVD are presented in line with K-means clustering for VQ. It is shown that MOD simplifies to K-means, while K-SVD fails to simplify due to its principle of updating. As MOD does not need to update the sparse representation vector during dictionary update stage, it is compatible to any structured/constrained sparsity model such as K-means. However, since MOD is not sequential, a sequential generalization to K-means is proposed that avoids the difficulties of K-SVD. Computational complexity for all algorithms are derived, and MOD is shown to be the least complex followed by SGK under a dimensionality condition, which is true for many practical application. Experimental results show that all the algorithms are performing equally well with marginal differences. Thus, MOD being the fastest among all, remains the dictionary training algorithm of choice for any kind of sparse representation. However, if sequential update becomes essential, SGK should be chosen.

References

[1] M. Aharon, M. Elad, and A. Bruckstein, "k -svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.

[2] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: theory and design," *Signal Processing*, vol. 80, no. 10, pp. 2121 – 2140, 2000.

[3] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[4] B. N. Datta, *Numerical Linear Algebra and Applications, Second Edition*. SIAM, 2010.

[5] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," Tech. Rep., Apr. 2008.

[6] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655 –4666, dec. 2007.