

# A Convergence Analysis of Distributed Dictionary Learning Based on the $K$ -SVD Algorithm

Haroon Raja and Waheed U. Bajwa

**Abstract**—This paper provides a convergence analysis of a recent distributed algorithm, termed cloud K-SVD, that solves the problem of data-adaptive representations for big, distributed data. It is assumed that a number of geographically-distributed, interconnected sites have massive local data and they are collaboratively learning a sparsifying dictionary underlying these data using cloud K-SVD. This paper provides a rigorous analysis of cloud K-SVD that gives insights into its properties as well as deviations of the dictionaries learned at individual sites from a centralized solution in terms of different measures of local/global data and topology of the interconnections.

## I. INTRODUCTION

Data representation using dictionary learning has gained a lot of attention in recent years. Some important contributions towards solving the dictionary learning problem include [1]–[3]. But such methods assume data to be present at a centralized location and are therefore not suited for cases when data are distributed across multiple locations. On the other hand, distributed data sets are quite prevalent in today’s information processing landscape. In order to address the challenge of dictionary learning from distributed data, [4]–[7] have recently proposed a few approaches. Among these approaches is a distributed variant of the efficient K-SVD algorithm for dictionary learning, termed cloud K-SVD [5]. Computationally, cloud K-SVD has been shown to have many of the desirable characteristics, such as fast convergence and small approximation error, of a dictionary learning algorithm [5]. Our goal in this paper is to provide a convergence analysis of cloud K-SVD.

In terms of our main contribution, note that cloud K-SVD relies on the power method for computing dominant singular vectors during the dictionary update step of K-SVD, while it uses consensus averaging to perform the power method in a distributed manner. In [5], we provided a preliminary analysis of cloud K-SVD that dealt with the convergence of its distributed power method component. In this paper we build upon our initial analysis in [5] and show that the cloud K-SVD dictionaries converge to the centralized  $K$ -SVD dictionary under certain assumptions. The implication of our analysis is that the differences between cloud K-SVD dictionaries and the centralized K-SVD dictionary can be arbitrarily small as long as both algorithms are initialized identically and appropriate numbers of power method and consensus iterations are performed in cloud K-SVD. Furthermore, our analysis guarantees this as long as total number of transmissions by any given site scales as  $\Omega(\log^2 S_{\max})$ , where  $S_{\max}$  denotes

the maximum number of data samples at any one site within the network.

To the best of our knowledge, this is the first work showing that dictionaries learned in distributed settings can be arbitrarily close to a centralized dictionary. While related works [4], [6] provide algorithms for distributed dictionary learning, they lack convergence guarantees. Other contributions like [7] focus on learning segments of the dictionary at each site, which is different from our setup since we are learning a complete dictionary at each site.

**Notation.** We use lower-case letters to represent scalars and vectors, while we use upper-case letters to represent matrices. Given a vector  $v$ ,  $\text{supp}(v)$  returns indices of the nonzero entries in  $v$ ,  $\|v\|_p$  denotes its  $\ell_p$  norm,  $\|v\|_0$  counts the number of its nonzero entries, and superscript  $(\cdot)^T$  denotes the transpose operation. Given a set  $\mathcal{I}$ ,  $v_{|\mathcal{I}}$  and  $A_{|\mathcal{I}}$  denote a subvector and a submatrix obtained by retaining entries of vector  $v$  and columns of matrix  $A$  corresponding to the indices in  $\mathcal{I}$ , respectively. Given matrices  $\{A_i \in \mathbb{R}^{n_i \times m_i}\}_{i=1}^N$ , the operation  $\text{diag}\{A_1, \dots, A_N\}$  returns a block-diagonal matrix  $A \in \mathbb{R}^{\sum n_i \times \sum m_i}$  that has  $A_i$ ’s on its diagonal. Finally, given a matrix  $A$ ,  $a_j$  and  $a_{j,T}$  denote the  $j^{\text{th}}$  column and the  $j^{\text{th}}$  row of  $A$ , respectively.

## II. PROBLEM FORMULATION

Our focus in this paper is on the convergence behavior of the cloud K-SVD algorithm [5]. Here, convergence of cloud K-SVD means that after  $T_d$  dictionary learning iterations, the gaps between the dictionaries using cloud K-SVD and the one learned using centralized K-SVD can be made arbitrarily small. Our goal in this regard is understanding the gaps between cloud K-SVD dictionaries and centralized K-SVD dictionary in terms of various problem parameters, such as the number of sites and data samples, the topology of interconnections, and the numbers of consensus and power method iterations.

### A. Collaborative dictionary learning using cloud K-SVD

Consider a collection of  $N$  geographically-distributed sites that are interconnected to each other according to a fixed topology. Mathematically, we represent this collection and their interconnections through an undirected graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N} = \{1, 2, \dots, N\}$  denotes the sites and  $\mathcal{E}$  denotes edges in the graph with  $(i, i) \in \mathcal{E}$  and  $(i, j) \in \mathcal{E}$  whenever there is a connection between site  $i$  and site  $j$ . The only assumption we make about the topology of  $\mathcal{G}$  is that it is a connected graph. Next, we assume that each site  $i$  has a massive collection of local data, expressed as a matrix  $Y_i \in \mathbb{R}^{n \times S_i}$  with  $S_i \gg n$  representing the number of data samples at the  $i^{\text{th}}$  site. We can express this distributed data as a single matrix  $Y = [Y_1 \ Y_2 \ \dots \ Y_N] \in \mathbb{R}^{n \times S}$ , where  $S = \sum_{i=1}^N S_i$  denotes the total number of data samples

This work is supported in part by the Army Research Office under grant W911NF-14-1-0295. The authors are with the Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08854 (Emails: haroon.raja@rutgers.edu and waheed.bajwa@rutgers.edu).

distributed across the  $N$  sites. In this setting, the fundamental objective is for each site to collaboratively learn a dictionary that underlies the global (distributed) data  $Y$ .

Assuming that global data  $Y$  is available at a centralized location, the dictionary learning problem can be expressed as

$$(D, X) = \arg \min_{D, X} \|Y - DX\|_F^2 \text{ s.t. } \forall s, \|x_s\|_0 \leq T_0, \quad (1)$$

where  $D \in \mathbb{R}^{n \times K}$  with  $K > n$  is an overcomplete dictionary having unit  $\ell_2$ -norm columns,  $X \in \mathbb{R}^{K \times S}$  corresponds to representation coefficients of the data having no more than  $T_0 \ll n$  nonzero coefficients per sample, and  $x_s, s = 1, \dots, S$  denotes the  $s^{\text{th}}$  column in  $X$ . Unlike classical dictionary learning, however, we do not have the global data  $Y$  available at a centralized location. Therefore, our goal is to have individual sites collaboratively learn dictionaries  $\{\hat{D}_i\}_{i \in \mathcal{N}}$  from global data  $Y$  such that these *collaborative dictionaries* are close to a dictionary  $D$  that could have been learned from  $Y$  in a centralized fashion.

Cloud K-SVD, given as Algorithm 1, was developed in [5] to accomplish the goal of collaborative dictionary learning from distributed data. This algorithm is a distributed variant of the famous K-SVD algorithm, which consists of two main steps, *sparse coding* and *dictionary update*. The sparse coding step in cloud K-SVD is performed only for locally available data at each site, while the dictionary update step is performed in a distributed manner using distributed power method. This makes the dictionary update step in cloud K-SVD as the most important and challenging step. Specifically, we recall from the description of K-SVD in [2] that dictionary update involves singular value decomposition (SVD) of the error matrix  $E_k^{(t)} = [E_{1,k}^{(t)} \dots E_{N,k}^{(t)}]$  in iteration  $t$ , which is available to K-SVD at one location. Cloud K-SVD, on the other hand, can only compute  $\hat{E}_{i,k}^{(t)}$  at any site  $i$  due to local sparse coding (cf. Step 5 of Algorithm 1), where  $\hat{E}_{i,k}^{(t)}$  denotes a perturbed version of the centralized  $E_{i,k}^{(t)}$ . Next, define an ordered set  $\tilde{\omega}_{i,k}^{(t)} = \{s : 1 \leq s \leq S, \tilde{x}_{i,k,T}^{(t)}(s) \neq 0\}$ , where  $\tilde{x}_{i,k,T}^{(t)}(s)$  denotes the  $s^{\text{th}}$  element of  $\tilde{x}_{i,k,T}^{(t)}$ , and an  $S \times |\tilde{\omega}_{i,k}^{(t)}|$  binary matrix  $\tilde{\Omega}_{i,k}^{(t)}$  that has ones at  $(\tilde{\omega}_{i,k}^{(t)}(s), s)$  locations and zeros everywhere else. Then each site  $i$  in cloud K-SVD only has access to  $\hat{E}_{i,k,R}^{(t)} = \hat{E}_{i,k}^{(t)} \tilde{\Omega}_{i,k}^{(t)}$  and  $\tilde{x}_{i,k,R}^{(t)} = \tilde{x}_{i,k,T}^{(t)} \tilde{\Omega}_{i,k}^{(t)}$ , whereas the centralized K-SVD has access to  $E_{i,k,R}^{(t)} = E_{i,k}^{(t)} \Omega_{i,k}^{(t)}$  and  $x_{i,k,R}^{(t)} = x_{i,k,T}^{(t)} \Omega_{i,k}^{(t)}$ . Finally, each site  $i$  in cloud K-SVD can only rely on  $\tilde{\Omega}_{i,k}^{(t)}$ , while K-SVD has access to the matrices  $\Omega_k^{(t)} = \text{diag}(\Omega_{1,k}^{(t)}, \dots, \Omega_{N,k}^{(t)})$  and  $E_{k,R}^{(t)} = E_k^{(t)} \Omega_k^{(t)}$ . Steps 4–21 in Algorithm 1 are designed to address these limitations of distributed dictionary learning; we refer the reader to [8] for further details on these steps.

### B. Main challenge

Even after K-SVD and cloud K-SVD are identically initialized,  $D^{(0)} = \hat{D}_i^{(0)}$ , we will have  $D^{(1)} \neq \hat{D}_i^{(1)}$  at the end of iteration 1 due to finite power method and consensus iterations in cloud K-SVD. This error in  $\hat{D}_i^{(1)}$  will then cause errors in sparse coding (Step 3 of Algorithm 1). Next, it can be seen from Steps 5–6 of Algorithm 1 that errors in sparse

---

### Algorithm 1: Cloud K-SVD for dictionary learning

---

**Input:** Local data  $Y_1, Y_2, \dots, Y_N$ , problem parameters  $K$  and  $T_0$ , and doubly-stochastic consensus matrix  $W$ .

**Initialize:** Generate  $d^{ref} \in \mathbb{R}^n$  and  $D^{init} \in \mathbb{R}^{n \times K}$  randomly, set  $t \leftarrow 0$  and  $\hat{D}_i^{(t)} \leftarrow D^{init}, i = 1, \dots, N$ .

```

1: while stopping rule do
2:    $t \leftarrow t + 1$ 
3:   (Sparse Coding) The  $i^{\text{th}}$  site solves
      $\forall s, \tilde{x}_{i,s}^{(t)} \leftarrow \arg \min_{x \in \mathbb{R}^K} \|y_{i,s} - \hat{D}_i^{(t-1)} x\|_2^2 \text{ s.t. } \|x\|_0 \leq T_0$ 
4:   for  $k = 1$  to  $K$  (Dictionary Update) do
5:      $\hat{E}_{i,k}^{(t)} \leftarrow Y_i - \sum_{j=1}^{k-1} \hat{d}_{i,j}^{(t)} \tilde{x}_{i,j,T}^{(t)} - \sum_{j=k+1}^K \hat{d}_{i,j}^{(t-1)} \tilde{x}_{i,j,T}^{(t)}$ 
6:      $\hat{E}_{i,k,R}^{(t)} \leftarrow \hat{E}_{i,k}^{(t)} \tilde{\Omega}_{i,k}^{(t)}$ 
7:      $\hat{M}_i^{(t)} \leftarrow \hat{E}_{i,k,R}^{(t)} \hat{E}_{i,k,R}^{(t)\top}$ 
8:     (Initialize Distributed Power Method) Generate
        $q^{init}$  randomly, set  $t_p \leftarrow 0$  and  $\hat{q}_i^{(t_p)} \leftarrow q^{init}$ 
9:     while stopping rule do
10:       $t_p \leftarrow t_p + 1$ 
11:      (Initialize Consensus Averaging) Set  $t_c \leftarrow 0$  and
         $z_i^{(t_c)} \leftarrow \hat{M}_i \hat{q}_i^{(t_p-1)}$ 
12:      while stopping rule do
13:         $t_c \leftarrow t_c + 1$ 
14:         $z_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} z_j^{(t_c-1)}$ 
15:      end while
16:       $\hat{v}_i^{(t_p)} \leftarrow z_i^{(t_c)} / [W_1^{t_c}]_i$ 
17:       $\hat{q}_i^{(t_p)} \leftarrow \hat{v}_i^{(t_p)} / \|\hat{v}_i^{(t_p)}\|_2$ 
18:    end while
19:     $\hat{d}_{i,k}^{(t)} \leftarrow \text{sgn}(\langle d^{ref}, \hat{q}_i^{(t_p)} \rangle) \hat{q}_i^{(t_p)}$ 
20:     $\tilde{x}_{i,k,R}^{(t)} \leftarrow \hat{d}_{i,k}^{(t)\top} \hat{E}_{i,k,R}^{(t)}$ 
21:  end for
22: end while
Return:  $\hat{D}_i^{(t)}, i = 1, 2, \dots, N$ .
```

---

coding and  $\hat{D}_i^{(1)}$  will result in deviation of  $\hat{E}_{i,k,R}^{(2)}$  from the centralized  $E_{i,k,R}^{(2)}$ . This means that the updated  $k^{\text{th}}$  atom in iteration 2 will have an error due to perturbation of  $E_{i,k,R}^{(2)}$  and due to errors caused by finite numbers of consensus and power method iterations. All these errors will keep on accumulating in the same way for any iteration  $t > 2$ . In summary, the main sources of error in cloud K-SVD are as follows:

- 1) Error in sparse coding due to perturbed dictionaries at the start of any iteration  $t > 1$ .
- 2) Error in  $E_{k,R}^{(t)}$  due to errors in dictionaries from the previous iteration and errors in sparse codes in the current iteration. This error in  $E_{k,R}^{(t)}$  will result in an error during the dictionary update step even if there is no error in computing the principal eigenvector of  $\hat{E}_{k,R}^{(t)} \hat{E}_{k,R}^{(t)\top}$ .
- 3) Error in computing the principal eigenvector of  $\hat{E}_{k,R}^{(t)} \hat{E}_{k,R}^{(t)\top}$  due to finite numbers of power method and consensus iterations.

Our goal in this paper is to analyze how these errors are accumulating in each iteration and how to control these errors such that the errors in dictionaries  $\hat{D}_i^{(t)}$  stay below some threshold after  $T_d$  dictionary learning iterations.

### III. ANALYSIS OF CLOUD K-SVD

Analysis of cloud K-SVD requires an understanding of the behavior of its major components, which include sparse coding, dictionary update, and distributed power method within dictionary update. In addition, one also expects that the closeness of  $\widehat{D}_i$ 's to the centralized solution will be a function of certain properties of local/global data. We begin our analysis of cloud K-SVD by first stating some of these properties in terms of the centralized K-SVD solution.

#### A. Preliminaries

We will start by providing algorithmic specification of the sparse coding steps in both algorithms. While the sparse coding step as stated in Step 3 of Algorithm 1 has combinatorial complexity, various low-complexity computational approaches can be used to solve this step in practice. Our analysis in the following will be focused on the case when sparse coding in both cloud K-SVD and centralized K-SVD is carried out using the *lasso* [9]. Specifically, we assume sparse coding is carried out by solving

$$x_{i,s} = \arg \min_{x \in \mathbb{R}^K} \frac{1}{2} \|y_{i,s} - Dx\|_2^2 + \tau \|x\|_1 \quad (2)$$

with the regularization parameter  $\tau > 0$  selected in a way that  $\|x_{i,s}\|_0 \leq T_0 \ll n$ . This can be accomplished, for example, by making use of the *least angle regression* algorithm [10]. Note that the lasso also has a dual, constrained form, given by

$$x_{i,s} = \arg \min_{x \in \mathbb{R}^K} \frac{1}{2} \|y_{i,s} - Dx\|_2^2 \quad \text{s.t.} \quad \|x\|_1 \leq \eta, \quad (3)$$

and the solutions of (2) and (3) are identical for an appropriate  $\eta_\tau = \eta(\tau)$  [11].

We also assume identical centralized and distributed initializations, i.e.,  $\widehat{D}_i^{(0)} = D^{(0)}, i = 1, \dots, N$ , where  $D^{(t)}, t \geq 0$ , in the following denotes the centralized K-SVD dictionary estimate in the  $t^{\text{th}}$  iteration. Despite identical initialization, the cloud K-SVD dictionaries get perturbed in each iteration due to imperfect power method and consensus averaging. In order to ensure these perturbations do not cause the cloud K-SVD dictionaries to diverge from the centralized solution after  $T_d$  iterations, we need the dictionary estimates returned by centralized K-SVD in each iteration to satisfy the following properties.

**[P1]** Let  $x_{i,s}^{(t)}$  denote the solution of the lasso (i.e., (2)) for  $D = D^{(t-1)}$  and  $\tau = \tau^{(t)}, t = 1, \dots, T_d$ . Then there exists some  $C_1 > 0$  such that the following holds:

$$\min_{t,i,s,j \notin \text{supp}(x_{i,s}^{(t)})} \tau^{(t)} - \left| \langle d_j^{(t)}, y_{i,s} - D^{(t-1)} x_{i,s}^{(t)} \rangle \right| > C_1.$$

For collection  $\{\tau^{(t)}\}_{t=1}^{T_d}$ , we also define the smallest regularization parameter  $\tau_{\min} = \min_t \tau^{(t)}$ , and the largest dual parameter among the (dual) collection  $\{\eta_\tau^{(t)} = \eta(\tau^{(t)})\}_{t=1}^{T_d}$  as  $\eta_{\tau, \max} = \max_t \eta_\tau^{(t)}$ .

**[P2]** Define  $\Sigma_{T_0} = \{\mathcal{I} \subset \{1, \dots, K\} : |\mathcal{I}| = T_0\}$ . Then there exists some  $C'_2 > \frac{C_1^4 \tau_{\min}^2}{1936}$  such that the following holds,  $\min_{t=1, \dots, T_d, \mathcal{I} \in \Sigma_{T_0}} \sigma_{T_0} \left( D_{|\mathcal{I}}^{(t-1)} \right) \geq \sqrt{C'_2}$ , where  $\sigma_{T_0}(\cdot)$  denotes the  $T_0^{\text{th}}$  (ordered) singular value of a

matrix. In our analysis, we will be using the parameter  $C_2 = \left( \sqrt{C'_2} - \frac{C_1^2 \tau_{\min}}{44} \right)^2$ .

**[P3]** Let  $\lambda_{1,k}^{(t)} > \lambda_{2,k}^{(t)} \geq \dots \lambda_{n,k}^{(t)} \geq 0$  denote the eigenvalues of the centralized ‘‘reduced’’ matrix  $E_{k,R}^{(t)} E_{k,R}^{(t)\top}, k \in \{1, \dots, K\}$ , in the  $t^{\text{th}}$  iteration,  $t \in \{1, \dots, T_d\}$ . Then there exists some  $C'_3 < 1$  such that the following holds,  $\max_{t,k} \frac{\lambda_{2,k}^{(t)}}{\lambda_{1,k}^{(t)}} \leq C'_3$ . Now define  $C_3 = \max \left\{ 1, \frac{1}{\min_{t,k} \lambda_{1,k}^{(t)} (1 - C'_3)} \right\}$ , which we will use in our forthcoming analysis.

Properties P1 and P2 correspond to sufficient conditions for  $x_{i,s}^{(t)}$  to be a unique solution of (2) [12] and guarantee that the centralized K-SVD generates a unique collection of sparse codes in each dictionary learning iteration. Property P3, on the other hand, ensures that algorithms such as the power method can be used to compute the dominant eigenvector of  $E_{k,R}^{(t)} E_{k,R}^{(t)\top}$  in each dictionary learning iteration (Steps 8–18 in Algorithm 1) [13]. In addition to these properties, our final analytical result for cloud K-SVD will also be a function of a certain parameter of the centralized error matrices  $\{E_k^{(t)}\}_{k=1}^K$  generated by the centralized K-SVD in each iteration. We define this parameter as follows. Let  $E_{i,k}^{(t)}, i = 1, \dots, N$ , denote part of the centralized error matrix  $E_k^{(t)}$  associated with the data of the  $i^{\text{th}}$  site in the  $t^{\text{th}}$  iteration, i.e.,  $E_k^{(t)} = \begin{bmatrix} E_{1,k}^{(t)} & E_{2,k}^{(t)} & \dots & E_{N,k}^{(t)} \end{bmatrix}, k = 1, \dots, K, t = 1, \dots, T_d$ . Then  $C_4 = \max \left\{ 1, \max_{t,i,k} \|E_{i,k}^{(t)}\|_2 \right\}$ .

#### B. Main Result

We are now ready to state the main result of this paper. This result is given in terms of the  $\|\cdot\|_2$  norm mixing time,  $T_{mix}$ , of the Markov chain associated with the doubly-stochastic weight matrix  $W$  used for consensus averaging, defined as

$$T_{mix} = \max_{i=1, \dots, N} \inf_{t \in \mathbb{N}} \left\{ t : \|e_i^\top W^t - \frac{1}{N} \mathbf{1}^\top\|_2 \leq \frac{1}{2} \right\}. \quad (4)$$

Here,  $e_i \in \mathbb{R}^N$  denotes the  $i^{\text{th}}$  column of the identity matrix  $I_N$ . In the following, main convergence results for cloud K-SVD along with brief discussions are presented. For detailed proofs and discussions we refer the reader to [8].

**Theorem 1** (Stability of Cloud K-SVD Dictionaries). *Suppose cloud K-SVD (Algorithm 1) and (centralized) K-SVD are identically initialized and both of them carry out  $T_d$  dictionary learning iterations. In addition, assume cloud K-SVD carries out  $T_p$  power method iterations during the update of each atom and  $T_c$  consensus iterations during each power method iteration. Finally, assume the K-SVD algorithm satisfies properties P1–P3. Next, define*

$$\alpha = \max_{t,k} \sum_{i=1}^N \|\widehat{E}_{i,k}^{(t)} \widehat{E}_{i,k}^{(t)\top}\|_2, \quad \beta = \max_{t,t_p,k} \frac{1}{\|\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t_p)} q_{c,t,k}\|_2},$$

$$\gamma = \max_{t,k} \sqrt{\sum_{i=1}^N \|\widehat{E}_{i,k}^{(t)} \widehat{E}_{i,k}^{(t)\top}\|_F^2}, \quad \nu = \max_{t,k} \frac{\lambda_{2,k}^{(t)}}{\lambda_{1,k}^{(t)}},$$

$$\widehat{\theta}_k^{(t)} \in [0, \pi/2] \quad \text{as} \quad \widehat{\theta}_k^{(t)} = \arccos \left( \frac{\left| \langle u_{1,k}^{(t)}, q^{init} \rangle \right|}{\|u_{1,k}^{(t)}\|_2 \|q^{init}\|_2} \right),$$

$$\mu = \max \left\{ 1, \max_{k,t} \tan(\widehat{\theta}_k^{(t)}) \right\}, \quad \text{and} \quad \zeta =$$

$K\sqrt{2S_{\max}} \left( \frac{6\sqrt{KT_0}}{\tau_{\min}C_2} + \eta_{\tau,\max} \right)$ , where  $S_{\max} = \max_i S_i$ ,  $u_{1,k}^{(t)}$  is the dominant eigenvector of  $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)\top}$ ,  $\widehat{\lambda}_{1,k}^{(t)}$  and  $\widehat{\lambda}_{2,k}^{(t)}$  are first and second largest eigenvalues of  $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)\top}$ , respectively, and  $q_{c,t,k}^{(t_p)}$  denotes the iterates of a centralized power method initialized with  $q^{init}$  for estimation of the dominant eigenvector of  $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)\top}$ . Then, assuming  $\min_{t,k} |\langle u_{1,k}^{(t)}, q^{init} \rangle| > 0$ , and fixing any  $\epsilon \in \left(0, \min \left\{ (10\alpha^2\beta^2)^{-1/3T_p}, \left(\frac{1-\nu}{4}\right)^{1/3} \right\}\right)$  and  $\delta_d \in \left(0, \min \left\{ \frac{1}{\sqrt{2}}, \frac{C_1^2\tau_{\min}}{44\sqrt{2}K} \right\}\right)$ , we have

$$\max_{\substack{i=1,\dots,N \\ k=1,\dots,K}} \left\| \widehat{d}_{i,k}^{(T_d)} \widehat{d}_{i,k}^{(T_d)\top} - d_k^{(T_d)} d_k^{(T_d)\top} \right\|_2 \leq \delta_d \quad (5)$$

as long as the number of power method iterations  $T_p \geq \frac{2(T_d K - 2) \log(8C_3 C_4^2 N + 5) + (T_d - 1) \log(1 + \zeta) + \log(8C_3 C_4 \mu N \sqrt{\nu} \delta_d^{-1})}{\log[(\nu + 4\epsilon^3)^{-1}]}$  and the number of consensus iterations  $T_c = \Omega(T_p T_{mix} \log(2\alpha\beta\epsilon^{-1}) + T_{mix} \log(\alpha^{-1}\gamma\sqrt{N}))$ .

We now comment on the major implications of Theorem 1. First, the theorem establishes that the distributed dictionaries  $\{\widehat{D}_i^{(T_d)}\}$  can indeed remain arbitrarily close to the centralized dictionary  $D^{(T_d)}$  after  $T_d$  dictionary learning iterations (cf. (5)). Second, the theorem shows that this can happen as long as the number of distributed power method iterations  $T_p$  scale in a certain manner. In particular, Theorem 1 calls for this scaling to be at least linear in  $T_d K$  (modulo the  $\log N$  multiplication factor), which is the total number of SVDs that K-SVD needs to perform in  $T_d$  dictionary learning iterations. On the other hand,  $T_p$  need only scale logarithmically with  $S_{\max}$ , which is significant in the context of big data problems. Other main problem parameters that affect the scaling of  $T_p$  include  $T_0$ ,  $n$ , and  $\delta_d^{-1}$ , all of which enter the scaling relation in a logarithmic fashion. Finally, Theorem 1 dictates that the number of consensus iterations  $T_c$  should also scale at least linearly with  $T_p T_{mix}$  (modulo some log factors) for the main result to hold. In summary, Theorem 1 guarantees that the distributed dictionaries learned by cloud K-SVD can remain close to the centralized dictionary without requiring excessive numbers of power method and consensus averaging iterations.

We now provide a brief heuristic understanding of the roadmap needed to prove Theorem 1. In the first dictionary learning iteration ( $t = 1$ ), we have  $\{\widehat{D}_i^{(t-1)} \equiv D^{(t-1)}\}$  due to identical initializations. While this means both K-SVD and cloud K-SVD result in identical sparse codes for  $t = 1$ , the distributed dictionaries begin to deviate from the centralized dictionary after this step. The perturbations in  $\{\widehat{d}_{i,k}^{(1)}\}$  happen due to the finite numbers of power method and consensus averaging iterations for  $k = 1$ , whereas they happen for  $k > 1$  due to this reason as well as due to the earlier perturbations in  $\{\widehat{d}_{i,j}^{(1)}, \widehat{x}_{i,j,T}^{(1)}\}$ ,  $j < k$ . In subsequent dictionary learning iterations ( $t > 1$ ), therefore, cloud K-SVD starts with already perturbed distributed dictionaries  $\{\widehat{D}_i^{(t-1)}\}$ . This in turn also results in deviations of the sparse codes computed by K-SVD and cloud K-SVD, which then adds another source of perturbations in  $\{\widehat{d}_{i,k}^{(t)}\}$  during the dictionary update steps. To summarize, imperfect power method and consensus averaging in cloud K-SVD introduce errors in the top eigenvector es-

timates of (centralized)  $E_{1,R}^{(1)} E_{1,R}^{(1)\top}$  at individual sites, which then accumulate for  $(k, t) \neq (1, 1)$  to also cause errors in estimate  $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)\top}$  of the matrix  $E_{k,R}^{(t)} E_{k,R}^{(t)\top}$  available to cloud K-SVD. Collectively, these two sources of errors cause deviations of the distributed dictionaries from the centralized dictionary and the proof of Theorem 1 mainly relies on our ability to control these two sources of errors.

### C. Roadmap to Theorem 1

The first main result needed for the proof of Theorem 1 looks at the errors in the estimates of the dominant eigenvector  $u_1$  of an arbitrary symmetric matrix  $M = \sum_{i=1}^N M_i$  obtained at individual sites using imperfect power method and consensus averaging when the  $M_i$ 's are distributed across the  $N$  sites (cf. Algorithm 1). The following result effectively helps us control the errors in cloud K-SVD dictionaries due to Steps 8–18 in Algorithm 1.

**Theorem 2** (Stability of Distributed Power Method). *Consider any symmetric matrix  $M = \sum_{i=1}^N M_i$  with dominant eigenvector  $u_1$  and eigenvalues  $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$ . Suppose each  $M_i$ ,  $i = 1, \dots, N$ , is only available at the  $i^{\text{th}}$  site in our network and let  $\widehat{q}_i$  denote an estimate of  $u_1$  obtained at site  $i$  after  $T_p$  iterations of the distributed power method (Steps 8–18 in Algorithm 1). Next, define  $\alpha_p = \sum_{i=1}^N \|M_i\|_2$ ,  $\beta_p = \max_{t_p=1,\dots,T_p} \frac{1}{\|M q_c^{(t_p)}\|_2}$ , and  $\gamma_p = \sqrt{\sum_{i=1}^N \|M_i\|_F^2}$ , where  $q_c^{(t_p)}$  denotes the iterates of a centralized power method initialized with  $q^{init}$ . Then, fixing any  $\epsilon \in (0, (10\alpha_p^2\beta_p^2)^{-1/3T_p})$ , we have*

$$\max_{i=1,\dots,N} \left\| u_1 u_1^\top - \widehat{q}_i \widehat{q}_i^\top \right\|_2 \leq \tan(\theta) \left| \frac{\lambda_2}{\lambda_1} \right|^{T_p} + 4\epsilon^{3T_p}, \quad (6)$$

as long as  $|\langle u_1, q^{init} \rangle| > 0$  and the number of consensus iterations within each iteration of the distributed power method (Steps 11–15 in Algorithm 1) satisfies  $T_c = \Omega(T_p T_{mix} \log(2\alpha_p\beta_p\epsilon^{-1}) + T_{mix} \log(\alpha_p^{-1}\gamma_p\sqrt{N}))$ . Here,  $\theta$  denotes the angle between  $u_1$  and  $q^{init}$ , defined as  $\theta = \arccos(|\langle u_1, q^{init} \rangle| / (\|u_1\|_2 \|q^{init}\|_2))$ .

Proof of this theorem can be found in our earlier work [5]. Theorem 2 states that  $\widehat{q}_i \xrightarrow{T_p} \pm u_1$  at an exponential rate at each site as long as enough consensus iterations are performed in each iteration of the distributed power method. In the case of a finite number of distributed power method iterations, (6) in Theorem 2 tells us that the maximum error in estimates of the dominant eigenvector is bounded by the sum of two terms, with the first term due to finite number of power method iterations and the second term due to finite number of consensus iterations.

The second main result needed to prove Theorem 1 looks at the errors between individual blocks of the reduced distributed error matrix  $\widehat{E}_{k,R}^{(t)} = \left[ \widehat{E}_{1,k,R}^{(t)}, \widehat{E}_{2,k,R}^{(t)}, \dots, \widehat{E}_{N,k,R}^{(t)} \right]$  and the reduced centralized error matrix  $E_{k,R}^{(t)} = \left[ E_{1,k,R}^{(t)}, E_{2,k,R}^{(t)}, \dots, E_{N,k,R}^{(t)} \right]$  for  $k \in \{1, 2, \dots, K\}$  and  $t \in \{1, 2, \dots, T_d\}$ . This result helps us control the error in Step 6 of Algorithm 1 and, together with Theorem 2,

characterizes the major sources of errors in cloud K-SVD in relation to centralized K-SVD. The following theorem provides a bound on error in  $E_{i,k,R}^{(t)}$

**Theorem 3** (Perturbation in the matrix  $\widehat{E}_{i,k,R}^{(t)}$ ). *Recall the definitions of  $\Omega_k^{(t)}$  and  $\widetilde{\Omega}_{i,k}^{(t)}$  from Section II-A. Next, express  $\Omega_k^{(t)} = \text{diag}\{\Omega_{1,k}^{(t)}, \dots, \Omega_{N,k}^{(t)}\}$ , where  $\Omega_{i,k}^{(t)}$  corresponds to the data samples associated with the  $i^{\text{th}}$  site, and define  $B_{i,k,R}^{(t)} = \widehat{E}_{i,k,R}^{(t)} - E_{i,k,R}^{(t)}$ . Finally, let  $\zeta, \mu, \nu, \epsilon$ , and  $\delta_d$  be as in Theorem 1, define  $\varepsilon = \mu\nu T_p + 4\epsilon^{3T_p}$ , and assume  $\varepsilon \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^2(T_dK-2)}$ . Then, if we perform  $T_p$  power method iterations and  $T_c = \Omega(T_p T_{\text{mix}} \log(2\alpha\beta\epsilon^{-1}) + T_{\text{mix}} \log(\alpha^{-1}\gamma\sqrt{N}))$  consensus iterations in cloud K-SVD and assume P1–P3 hold, we have for  $i \in \{1, \dots, N\}$ ,  $t \in \{1, 2, \dots, T_d\}$ , and  $k \in \{1, 2, \dots, K\}$*

$$\|B_{i,k,R}^{(t)}\|_2 \leq \begin{cases} 0, & \text{for } t = 1, k = 1, \\ \varepsilon(1 + \zeta)^{t-1}C_4(8C_3NC_4^2 + 5)^{(t-1)K+k-2}, & \text{o.w.} \end{cases}$$

Theorem 3 tells us that the error in matrix  $E_{i,k,R}^{(t)}$  can be made arbitrarily small through a suitable choice of  $T_p$  and  $\epsilon$  as long as all of the assumptions of Theorem 1 are satisfied. One of the steps in proving Theorem 1 involves proving that the assumption on  $\varepsilon$  is satisfied as long as we are performing power method iterations and consensus iterations as required by Theorem 1 (see to [8] for complete proof). In the following, we provide a brief sketch of the proof of Theorem 3 and refer the reader to [8] for complete details.

We can prove Theorem 3 by induction over dictionary learning iteration  $t$ . But first we need to have a way to bound  $\|B_{i,k+1,R}^{(t)}\|_2$  using bounds on  $\{\|B_{i,j,R}^{(t)}\|_2\}_{j=1}^k$  and also we need to have a method to bound  $\|B_{i,1,R}^{(t+1)}\|_2$  using bounds on  $\{\|B_{i,j,R}^{(t)}\|_2\}_{j=1}^K$ . Notice from Algorithm 1 that sparse coding is always performed before update of the first dictionary atom. But we do not perform sparse coding before updating any other dictionary atom. Due to this distinction, we address the problem of error accumulation in matrix  $E_{i,k,R}^{(t)}$  for first dictionary atom ( $\|B_{i,1,R}^{(t+1)}\|_2$ ) differently than for any other dictionary atom ( $\{\|B_{i,j,R}^{(t)}\|_2\}_{j=2}^K$ ). Proof of Theorem 3 can then be divided into three steps.

**Bound on  $\|B_{i,k+1,R}^{(t)}\|_2$ :** Recall from Steps 5–6 in Algorithm 1 that  $\widehat{E}_{i,k,R}^{(t)} = Y_i \widetilde{\Omega}_{i,k}^{(t)} - \sum_{j=1}^{k-1} \widehat{d}_{i,j}^{(t)} \widehat{x}_{i,j,T}^{(t)} \widetilde{\Omega}_{i,k}^{(t)} - \sum_{j=k+1}^K \widehat{d}_{i,j}^{(t-1)} \widehat{x}_{i,j,T}^{(t-1)} \widetilde{\Omega}_{i,k}^{(t)}$ . Now, if one assumes that  $\widetilde{\Omega}_k^{(t)} = \Omega_k^{(t)}$ , which can be argued to be true using results from [14] and assumptions of Theorem 1, then the error in  $E_{i,k,R}^{(t)}$  is due to errors in  $\{\widehat{x}_{i,j,T}^{(t)}\}_{j=1}^K$  and  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ . Infact, to bound  $\|B_{i,k+1,R}^{(t)}\|_2$  we only need to know bounds on errors in  $d_{i,k}^{(t)}$  and  $x_{i,k,T}^{(t)}$ . Next, recall from Step 20 in Algorithm 1 that  $\widehat{x}_{i,k,R}^{(t)} = \widehat{d}_{i,k}^{(t)} \widehat{E}_{i,k,R}^{(t)}$ , which means we only need to know a bound on  $d_k^{(t)}$  to bound  $\|B_{i,k+1,R}^{(t)}\|_2$ . But the challenge is to bound error in  $d_k^{(t)}$  from a given bound on  $\|B_{i,k,R}^{(t)}\|_2$ . This is accomplished by noting that there are two sources of error in  $d_k^{(t)}$ . The first source is the difference in eigenvectors of  $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)\top}$  and  $E_{k,R}^{(t)} E_{k,R}^{(t)\top}$ . We will bound this difference

using [13, Theorem 8.1.12]. The second source of error in  $d_k^{(t)}$  is the error in eigenvector computation, which in our case is due to the distributed power method. It follows from Theorem 2 and statement of Theorem 3 that this error is bounded by  $\varepsilon$ . Combining these two sources of error, we can bound the error in  $d_k^{(t)}$ , which we use to bound  $\|B_{i,k+1,R}^{(t)}\|_2$ .

**Bound on  $\|B_{i,1,R}^{(t+1)}\|_2$ :** In order to bound  $\|B_{i,1,R}^{(t+1)}\|_2$  when we know bounds on  $\{\|B_{i,j,R}^{(t)}\|_2\}_{j=1}^K$ , the difference from previous case is that now we can not write sparse code  $\{\widehat{x}_{i,j,T}^{(t+1)}\}_{j=1}^K$  in terms of dictionary atoms  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ . Therefore, in addition to bounding errors in dictionary atoms  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ , we also need to bound errors in sparse codes due to perturbations in dictionaries after iteration  $t$ . Since we know  $\{\|B_{i,k,R}^{(t)}\|_2\}_{j=1}^K$ , we can use these to bound  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ . Next, using error bounds on  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ , we can use [14, Theorem 4] to bound errors in  $\{\widehat{x}_{i,j,T}^{(t+1)}\}_{j=1}^K$ . Finally, using these error bounds on  $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$  and  $\{\widehat{x}_{i,j,T}^{(t+1)}\}_{j=1}^K$  we can bound  $\|B_{i,1,R}^{(t+1)}\|_2$ .

**Bound on  $\|B_{i,k,R}^{(t)}\|_2, \forall t$  and  $k$ :** Next using induction argument over  $t$  we can prove Theorem 3.

#### IV. CONCLUSION

In this paper, we have provided mathematical analysis of cloud K-SVD. Our analysis shows that under certain assumptions if we perform enough numbers of power method and consensus iterations then the cloud K-SVD dictionaries converge to the centralized K-SVD solution.

#### REFERENCES

- [1] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, Feb. 2003.
- [2] 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, Nov. 2006.
- [3] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *JMLR*, vol. 11, pp. 19–60, 2010.
- [4] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2013.
- [5] H. Raja and W. U. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Proc. 51st Annual Allerton Conference on Communication, Control, and Computing*, 2013, pp. 1474–1481.
- [6] L. J. Z. M, Z. X, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Trans. on Image Processing*, vol. 23, no. 6, pp. 2528–2541, 2014.
- [7] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Trans. Signal Processing*, vol. 63, no. 4, pp. 1001 – 1016, Feb. 2015.
- [8] H. Raja and W. U. Bajwa, "Cloud K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *arXiv preprint arXiv:1412.7839*, 2014.
- [9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996.
- [10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–451, 2004.
- [11] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Select. Topics Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
- [12] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1341–1344, Jun. 2004.
- [13] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 2012.
- [14] N. Mehta and A. G. Gray, "Sparsity-based generalization bounds for predictive sparse coding," in *Proc. 30th Intl. Conf. Machine Learning (ICML'13)*, Atlanta, GA, Jun. 2013, pp. 36–44.