Cloud K-SVD: A Collaborative Dictionary Learning Algorithm for Big, Distributed Data

Haroon Raja and Waheed U. Bajwa

***Abstract*—This paper studies 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 interested in collaboratively learning a low-dimensional geometric structure underlying these data. In contrast to previous works on subspace-based data representations, this paper focuses on the geometric structure of a union of subspaces (UoS). In this regard, it proposes a distributed algorithm—termed cloud K-SVD—for collaborative learning of a UoS structure underlying distributed data of interest. The goal of cloud K-SVD is to learn a common overcomplete dictionary at each individual site such that every sample in the distributed data can be represented through a small number of atoms of the learned dictionary. Cloud K-SVD accomplishes this goal without requiring exchange of individual samples between sites. This makes it suitable for applications where sharing of raw data is discouraged due to either privacy concerns or large volumes of data. This paper also provides an 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 interconnections. Finally, the paper numerically illustrates the efficacy of cloud K-SVD on real and synthetic distributed data.**

***Index Terms*—Consensus averaging, dictionary learning, distributed data, K-SVD, power method, sparse coding.**

I. Introduction

Modern information processing is based on the axiom that while real-world data may live in high-dimensional ambient spaces, relevant information within them almost always lies near low-dimensional geometric structures. Knowledge of these (low-dimensional) geometric structures underlying data of interest is central to the success of a multitude of information processing tasks. But this knowledge is unavailable to us in an overwhelmingly large number of applications and a great deal of work has been done in the past to *learn* geometric structure of data from the data *themselves*. Much of that work, often studied under rubrics such as *principal component analysis* (PCA) [3], *generalized PCA* [4], *hybrid linear modeling* [5], and *dictionary learning* [6]–[8], has been focused on centralized settings in which the entire data are assumed available at a single location. In recent years, there has been some effort to extend these works to distributed settings; see, e.g., [9]–[20]. The setup considered in some of these works is that each distributed entity is responsible for either some dimensions of the data [9]–[11] or some part of the learned geometric structure [10], [11], [16]. Other works in this direction also focus on learning under the assumption of data lying near *(linear) subspaces* [9]–[13], require extensive communications among the distributed entities [14], and ignore some of the technical details associated with processing among distributed entities having interconnections described by graphs of arbitrary, unknown topologies [12]–[15].

In this paper, we are interested in a setting in which a number of geographically-distributed sites have massive local data and these sites are interested in collaboratively learning a geometric structure underlying their data by communicating among themselves over public/private networks. The key constraints in this problem that distinguish it from some of the prior works are: (i) sites cannot communicate "raw" data among themselves; (ii) interconnections among sites are not described by a complete graph; and (iii) sites do not have knowledge of the global network topology. All these constraints are reflective of the future of big, distributed data in the world. In particular, the first constraint is justified because of the size of local data compilations as well as privacy concerns in the modern age. Similarly, the latter two constraints are justified because linking geographically-distributed sites into a complete graph can be cost prohibitive and since enterprises tend to be protective of their internal network topologies.

A. Our Contributions

The first main contribution of this paper is formulation of a distributed method, which we term as *cloud K-SVD*, that enables data-adaptive representations in distributed settings. In contrast to works that assume a linear geometric structure for data [9]–[13], cloud K-SVD is based on the premise that data lie near a *union* of low-dimensional subspaces. The *union-of-subspaces* (UoS) model is a nonlinear generalization of the subspace model [21] and has received widespread acceptance in the community lately. The task of learning the UoS underlying data of interest from data themselves is often termed *dictionary learning* [6]–[8], which involves data-driven learning of an overcomplete dictionary such that every data sample can be approximated through a small number of atoms of the dictionary. Dictionary learning—when compared to linear data-adaptive representations such as the PCA and the linear discriminant analysis [22]—has been shown to be highly effective for tasks such as compression [6], denoising [23], object recognition [24], and inpainting [25]. Cloud K-SVD, as the name implies, is a distributed variant of the popular dictionary learning algorithm K-SVD [7] and leverages a

This work is supported in part by the ARO under grant W911NF-14-1-0295 and by the NSF under grants CCF-1453073 and CCF-1525276. Some of the results reported here were presented at Allerton Conf. on Communication, Control, and Computing, 2013 [1] and Intl. Symp. on Information Theory, 2015 [2]. The authors are with the Department of Electrical and Computer Engineering, Rutgers University–New Brunswick (Emails: haroon.raja@rutgers.edu and waheed.bajwa@rutgers.edu).

classical iterative eigenvector estimation algorithm, termed the *power method* [26, Ch. 8], and consensus averaging [27] for collaborative dictionary learning.

The second main contribution of this paper is 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 the centralized K-SVD solution in terms of different measures of local/global data and topology of the interconnections. Using tools from linear algebra, convex optimization, matrix perturbation theory, etc., our analysis shows that—under identical initializations—the dictionaries learned by cloud K-SVD come arbitrarily close to the one learned by (centralized) K-SVD as long as appropriate number of power method and consensus iterations are performed in each iteration of cloud K-SVD. Finally, the third main contribution of this paper involves numerical experiments on synthetic and real data that demonstrate both the efficacy of cloud K-SVD and the usefulness of collaborative dictionary learning over local dictionary learning.

B. Relationship to Previous Work

Some of the earliest works in distributed processing date back nearly three decades [28], [29]. Since then a number of distributed methods have been proposed for myriad tasks. Some recent examples of this that do not involve a centralized *fusion center* include distributed methods for classification [30]–[32], localization [33], [34], linear regression [35], and (multitask) estimation [19], [20], [36]. But relatively little attention has been paid to the problem of data-driven distributed learning of the geometric structure of data. Notable exceptions to this include [11]–[18]. While our work as well as [11]–[15] rely on consensus averaging for computing the underlying geometric structure, we are explicit in our formulation that perfect consensus under arbitrary, unknown topologies cannot be achieved. In contrast, developments in [11]–[15] are carried out under the assumption of infinite-precision consensus averaging. Further, [11]–[13] assume a subspace data model, while [14] advocates the use of consensus averaging for computing sample covariance—an approach that requires extensive communications among the distributed entities.

Our work is most closely related to that in [16]–[18], which also study dictionary learning in distributed settings. But [16] focuses only on learning parts of the dictionary at each site as opposed to the setup of this paper in which we are interested in learning a complete dictionary at each site. While this paper and [17], [18] share the same setup, our work as well as [18] are fundamentally different from [17]. The method proposed in [17] involves learning local dictionaries at different sites and then *diffusing* these local dictionaries to obtain a global dictionary. In contrast, our work and [18] are based on the centralized K-SVD algorithm, which is known to be superior to other dictionary learning methods [7], and involve updating each atom of the local dictionaries in a collaborative fashion. The difference between this work and [18] lies in the fact that cloud K-SVD uses a distributed variant of the power method to update each atom, whereas [18] relies on distributed optimization for this purpose. This helps us rigorously analyze the performance of cloud K-SVD, whereas no such analysis is provided in [18]. Note that while we analyzed the distributed power method component of cloud K-SVD in our earlier work [1], this paper extends that work to provide a comprehensive analysis of the entire algorithm.

We conclude by noting that the distributed power method component of cloud K-SVD has similarities with the work in [12], [37]. However, unlike [12], [37], we do not assume perfect consensus during iterations of the power method, which leaves open the question of convergence of the distributed variant of the power method. While analyzing cloud K-SVD, we in fact end up addressing this question also. That part of our analysis is reminiscent of the one carried out in [38] in the context of convergence behavior of distributed eigenanalysis of a network using a power method-like iterative algorithm. However, there are fundamental differences in the analysis of [38] and our work because of the exact place where consensus averaging is carried out in the two works, which is dictated by the distinct nature of the two applications.

C. Notation and Paper Organization

We use lower-case letters to represent scalars and vectors, while we use upper-case letters to represent matrices. The operator sgn : $\mathbb{R} \to \{+1, -1\}$ is defined as $\text{sgn}(x) = x/|x|$, while $\text{supp}(v)$ returns indices of the nonzero entries in vector v. Superscript $(\cdot)^\mathsf{T}$ denotes the transpose operation, $\|\cdot\|_0$ counts the number of nonzero entries in a vector, $\|v\|_p$ denotes the usual ℓ_p norm of vector v, and $\langle u, v\rangle$ denotes the inner product between vectors u and v. 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, while $\|A\|_2$, $\|A\|_F$, and $\|A\|_{\max}$ denote the operator norm, Frobenius norm, and max norm (i.e., maximum absolute value) of matrix A, respectively. Given matrices $\{A_i \in \mathbb{R}^{n_i \times m_i}\}_{i=1}^N$, $\text{diag}\{A_1, \ldots, A_N\}$ denotes 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^{th} column and the j^{th} row of A, respectively.

The rest of this paper is organized as follows. In Sec. II, we formulate the problem of collaborative dictionary learning from big, distributed data. In Sec. III, we describe the cloud K-SVD algorithm. In Sec. IV, we provide an analysis of cloud K-SVD algorithm. We provide some numerical results in Sec. V and concluding remarks in Sec. VI. Finally, proofs of main theorems stated in Sec. IV are given in appendices.

II. Problem Formulation

In this paper, we consider a collection of N geographically-distributed sites that are interconnected to each other according to a fixed topology. Here, we use "site" in the broadest possible sense of the term, with a site corresponding to a single computational system (e.g., sensor, drone, smartphone, tablet, server, database), a collection of co-located computational systems (e.g., data center, computer cluster, robot swarm), etc. Mathematically, we represent this collection and their interconnections through an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \cdots, N\}$ denotes the sites and $\mathcal{E}$ denotes

edges in $\mathcal{G}$ with $(i,i) \in \mathcal{E}$, while $(i,j) \in \mathcal{E}$ whenever there is a connection between sites i and j. The only assumption we make about the topology of $\mathcal{G}$ is that it is a connected graph.

Next, we assume each site i has a collection of local data, expressed as a matrix $Y_i \in \mathbb{R}^{n \times S_i}$ with S_i representing the number of data samples at the i^{th} site. We can express all this distributed data into a single matrix $Y = \begin{bmatrix} Y_1 & \dots & Y_N \end{bmatrix} \in \mathbb{R}^{n \times S}$, where $S = \sum_{i=1}^{N} S_i$ denotes the total number of data samples distributed across the N sites; see Fig. 1 for a schematic representation of this. In this setting, the fundamental objective is for each site to collaboratively learn a low-dimensional geometric structure that underlies the global (distributed) data Y. The basic premises behind collaborative structure learning of global data, as opposed to local structure learning of local data, are manifold. First, since the number of global samples is much larger than the number of local samples, we expect that collaborative learning will outperform local learning for data representations. Second, local learning will be strictly suboptimal for some sites in cases where sampling density, noise level, fraction of outliers, etc., are not uniform across all sites. Collaborative learning, on the other hand, will even out such nonuniformities within local data.

Our main assumption is that the low-dimensional geometric structure underlying the global data corresponds to a union of T_0-dimensional subspaces in $\mathbb{R}^n$, where $T_0 \ll n$. One possible means of learning such a structure is studied under the moniker dictionary learning, which learns an *overcomplete dictionary* D such that each data sample is well approximated by no more than T_0 columns (i.e., *atoms*) of D [6]–[8]. Assuming the global data Y is available at a centralized location, this problem of dictionary learning can be expressed as

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

where $D \in \mathbb{R}^{n \times K}$ with $K > n$ is an overcomplete dictionary having unit ℓ_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 denotes the s^{th} column in X. Note that (1) is non-convex in $\big(D, X\big)$, although it is convex in D alone. One of the most popular approaches to solving (1) involves alternate minimization in which one alternates between solving (1) for D using a fixed X and then solving (1) for X using a fixed D [7], [39].

Unlike classical dictionary learning, however, we do not have the global data Y available at a centralized location. Data aggregation either at a centralized location or at any one of the individual sites is also impractical due to communications and storage costs of big data. Furthermore, privacy issues may also preclude aggregation of data. Instead, our goal is to have individual sites collaboratively learn dictionaries $\{\widehat{D}_i\}_{i \in \mathcal{N}}$ 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. In the following section, we present a distributed variant of a popular dictionary learning algorithm that accomplishes this goal without exchanging raw data between sites. This is followed by a rigorous analysis of the proposed algorithm in Sec. IV, which establishes that the collaborative dictionaries learned using our proposed algorithm can indeed be made to come arbitrarily close to a centralized dictionary.

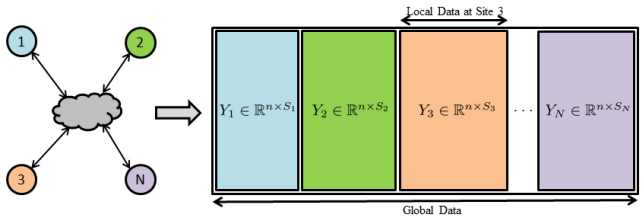


Fig. 1. A schematic representing global data Y distributed across N sites. Here, n denotes the dimension of each data sample, while S_i denotes the total number of data samples available at the i^{th} site.

III. Cloud K-SVD

In this paper, we focus on the K-SVD algorithm [7] as the basis for collaborative dictionary learning. We have chosen to work with K-SVD because of its iterative nature and its reliance on the singular value decomposition (SVD), both of which enable its exploitation for distributed purposes. In the following, we first provide a brief overview of K-SVD, which is followed by presentation of our proposed algorithm—termed cloud K-SVD—for collaborative dictionary learning.

A. Dictionary Learning Using K-SVD

The K-SVD algorithm initializes with a (often randomized) dictionary $D^{(0)}$ and solves (1) by iterating between two stages: a *sparse coding stage* and a *dictionary update stage* [7]. Specifically, for a fixed estimate of the dictionary $D^{(t-1)}$ at the start of iteration $t \geq 1$, the sparse coding stage in K-SVD involves solving (1) for $X^{(t)}$ as follows:

$$\forall s, \; x_s^{(t)} = \arg\min_{x \in \mathbb{R}^K} \|y_s - D^{(t-1)}x\|_2^2 \text{ s.t. } \|x\|_0 \leq T_0, \quad (2)$$

where y_s denotes the s^{th} column of Y. While (2) in its stated form has combinatorial complexity, it can be solved by either convexifying (2) [40] or using greedy algorithms [41].

After the sparse coding stage, K-SVD fixes $X^{(t)}$ and moves to the dictionary update stage. The main novelty in K-SVD lies in the manner in which it carries out dictionary update, which involves iterating through the K atoms of $D^{(t-1)}$ and individually updating the k^{th} atom, $k \in 1, \dots, K$, as follows:

$$d_k^{(t)} = \arg\min_{d \in \mathbb{R}^n} \left\| \Big(Y - \sum_{j=1}^{k-1} d_j^{(t)} x_{j,T}^{(t)} - \sum_{j=k+1}^{K} d_j^{(t-1)} x_{j,T}^{(t)}\Big) - d\, x_{k,T}^{(t)} \right\|_F^2 = \arg\min_{d \in \mathbb{R}^n} \left\| E_k^{(t)} - d\, x_{k,T}^{(t)} \right\|_F^2. \quad (3)$$

Here, $E_k^{(t)}$ is the representation error for Y using first $k-1$ atoms of $D^{(t)}$ and last $k+1, \dots, K$ atoms of $D^{(t-1)}$. In order to simplify computations, K-SVD in [7] further defines an ordered set $\omega_k^{(t)} = \{s : 1 \leq s \leq S, x_{k,T}^{(t)}(s) \neq 0\}$, where $x_{k,T}^{(t)}(s)$ denotes the s^{th} element of $x_{k,T}^{(t)}$, and an $S \times |\omega_k^{(t)}|$ binary matrix $\Omega_k^{(t)}$ that has ones in $(\omega_k^{(t)}(s), s)$ locations and zeros everywhere else. Then, defining $E_{k,R}^{(t)} = E_k^{(t)} \Omega_k^{(t)}$ and $x_{k,R}^{(t)} = x_{k,T}^{(t)} \Omega_k^{(t)}$, it is easy to see from (3) that

$d_k^{(t)} = \arg\min_{d\in\mathbb{R}^n} \left\| E_{k,R}^{(t)} - d\, x_{k,R}^{(t)} \right\|_F^2$. Therefore, solving (3) is equivalent to finding the best rank-one approximation of $E_{k,R}^{(t)}$, which is given by the Eckart–Young theorem as $d_k^{(t)} x_{k,R}^{(t)} = \sigma_1 u_1 v_1^\mathsf{T}$, where u_1 and v_1 denote the largest left- and right-singular vectors of $E_{k,R}^{(t)}$, respectively, while σ_1 denotes the largest singular value of $E_{k,R}^{(t)}$. The k^{th} atom of $D^{(t)}$ can now simply be updated as $d_k^{(t)} = u_1$. It is further advocated in [7] that the k^{th} row of the "reduced" coefficient matrix, $x_{k,R}^{(t)}$, should be simultaneously updated to $x_{k,R}^{(t)} = \sigma_1 v_1^\mathsf{T}$. The dictionary update stage in K-SVD involves K such applications of the Eckart–Young theorem to update the K atoms of $D^{(t-1)}$ and the K "reduced" rows of $X^{(t)}$. The algorithm then moves to the sparse coding stage and continues alternating between the two stages till a stopping criterion (e.g., a prescribed representation error) is reached.

B. Collaborative Dictionary Learning Using Cloud K-SVD

We now present our collaborative dictionary learning algorithm based on K-SVD. The key to distributing K-SVD is understanding ways in which both the sparse coding and the dictionary update stages can be distributed. To this end, we assume collaborative dictionary learning is in iteration $t \geq 1$ and each site i in this iteration has a local estimate $\widehat{D}_i^{(t-1)}$ of the desired dictionary from the previous iteration. In order for the sparse coding stage to proceed, we propose that each site computes representation coefficients of its local data without collaborating with other sites by locally solving Step 3 of Algorithm 1, i.e.,

$$\forall s,\ \widetilde{x}_{i,s}^{(t)} = \arg\min_{x\in\mathbb{R}^K} \|y_{i,s} - \widehat{D}_i^{(t-1)} x\|_2^2 \text{ s.t. } \|x\|_0 \leq T_0, \quad (4)$$

where $y_{i,s}$ and $\widetilde{x}_{i,s}^{(t)}$ denote the s^{th} sample and its coefficient vector at site i, respectively. This "local" sparse coding for collaborative dictionary learning simplifies the sparse coding stage and is justified as long as the local dictionary estimates $\widehat{D}_i^{(t-1)}$ remain close to each other (established in Sec. IV).

The next challenge in collaborative dictionary learning based on K-SVD arises during the dictionary update stage. Recall that the dictionary update stage in K-SVD involves computing the largest left- and right-singular vectors of the "reduced" error matrix $E_{k,R}^{(t)} = E_k^{(t)} \Omega_k^{(t)}, k \in \{1,\ldots,K\}$. However, unless the local dictionary estimates $\widehat{D}_i^{(t-1)}$ happen to be identical, we end up with N such (reduced) error matrices in a distributed setting due to N different local dictionary estimates. To resolve this, we propose to use the following definition of the reduced error matrix in a distributed setting: $\widehat{E}_{k,R}^{(t)} = \begin{bmatrix} \widehat{E}_{1,k,R}^{(t)} & \ldots & \widehat{E}_{N,k,R}^{(t)} \end{bmatrix}$, where $\widehat{E}_{i,k,R}^{(t)} = Y_i \widetilde{\Omega}_{i,k}^{(t)} - \left(\sum_{j=1}^{k-1} \widehat{d}_{i,j}^{(t)} \widehat{x}_{i,j,T}^{(t)} + \sum_{j=k+1}^{K} \widehat{d}_{i,j}^{(t-1)} \widetilde{x}_{i,j,T}^{(t)} \right) \widetilde{\Omega}_{i,k}^{(t)}$. Here, $\widetilde{x}_{i,j,T}^{(t)}$ denotes the j^{th} row of coefficient matrix $\widetilde{X}_i^{(t)}$ computed at site i during the sparse coding step performed on Y_i using $\widehat{D}_i^{(t-1)}$ at the start of iteration t, while $\widehat{x}_{i,j,T}^{(t)}$ denotes the j^{th} row of the updated coefficient matrix $\widehat{X}_i^{(t)}$ available at site i due to the update in coefficient matrix performed during the dictionary update step. Furthermore, $\widetilde{\Omega}_{i,k}^{(t)}$ is similar to $\Omega_k^{(t)}$ defined for K-SVD except that it is now defined for only local coefficient matrix $\widetilde{X}_i^{(t)}$ at site i.

Next, in keeping with the K-SVD derivation in [7], we propose that each of the N sites updates the k^{th} atom of its respective local dictionary and the k^{th} row of its respective "reduced" coefficient matrix, $\widehat{x}_{i,k,R}^{(t)} = \widehat{x}_{i,k,T}^{(t)} \widetilde{\Omega}_{i,k}^{(t)}$, by collaboratively computing the dominant left- and right-singular vectors of the *distributed error matrix* $\widehat{E}_{k,R}^{(t)}$, denoted by u_1 and v_1, respectively.[1] In fact, since $u_1^\mathsf{T} \widehat{E}_{k,R}^{(t)} = \sigma_1 v_1$ with σ_1 being the largest singular value of $\widehat{E}_{k,R}^{(t)}$, it follows that if a site has access to the dominant left-singular vector, u_1, of $\widehat{E}_{k,R}^{(t)}$ then it can simply update the k^{th} row of its respective "reduced" coefficient matrix by setting $\widehat{d}_{i,k}^{(t)} = u_1$ and setting $\widehat{x}_{i,k,R}^{(t)} = \widehat{d}_{i,k}^{(t)^\mathsf{T}} \widehat{E}_{i,k,R}^{(t)}$. Therefore, we need only worry about collaborative computation of u_1 in this setting. To this end, we define $\widehat{M}^{(t)} = \widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^\mathsf{T}}$ and note that u_1 corresponds to the dominant eigenvector of $\widehat{M}^{(t)}$. Now express $\widehat{M}^{(t)}$ as $\widehat{M}^{(t)} = \sum_{i=1}^N \widehat{M}_i^{(t)}$ and notice that each $\widehat{M}_i^{(t)} = \widehat{E}_{i,k,R}^{(t)} \widehat{E}_{i,k,R}^{(t)^\mathsf{T}}$ is a matrix that is readily computable at each local site. Our goal now is computing the dominant eigenvector of $\widehat{M}^{(t)} = \sum_{i=1}^N \widehat{M}_i^{(t)}$ in a collaborative manner at each site. In order for this, we will make use of distributed power method, which has been invoked previously in [12], [13], [38] and which corresponds to a distributed variant of the classical power method for eigenanalysis [26].

Distributed Power Method: Power method is an iterative procedure for computing eigenvectors of a matrix. It is simple to implement and, assuming that the largest eigenvalue λ_1 of a matrix is strictly greater than its second-largest eigenvalue λ_2, it converges to the subspace spanned by the dominant eigenvector at an exponential rate. We are interested in a distributed variant of the power method to compute the dominant eigenvector of $\widehat{M}^{(t)} = \sum_{i=1}^N \widehat{M}_i^{(t)}$, where the $\widehat{M}_i^{(t)}$'s are distributed across N sites. To this end, we proceed as follows.

First, all sites initialize to the same (unit-norm) estimate of the eigenvector $\widehat{q}_i^{(0)} = q^{init}$.[2] Next, assuming that the sites are carrying out iteration t_p of the distributed power method, each site computes $\widehat{M}_i^{(t)} \widehat{q}_i^{(t_p-1)}$ locally, where $\widehat{q}_i^{(t_p-1)}$ denotes an estimate of the dominant eigenvector of $\widehat{M}^{(t)}$ at the i^{th} site after $t_p - 1$ power method iterations. In the next step, the sites collaboratively compute an approximation $\widehat{v}_i^{(t_p)}$ of $\sum_i \widehat{M}_i^{(t)} \widehat{q}_i^{(t_p-1)}$ at each site. In the final step of the t_p^{th} iteration of the distributed power method, each site normalizes its estimate of the dominant eigenvector of $\widehat{M}^{(t)}$ locally: $\widehat{q}_i^{(t_p)} = \widehat{v}_i^{(t_p)} / \|\widehat{v}_i^{(t_p)}\|_2$.

It is clear from the preceding discussion that the key in distributed power method is the ability of the sites to collaboratively compute an approximation of $\sum_i \widehat{M}_i^{(t)} \widehat{q}_i^{(t_p-1)}$ in each it-

[1] An alternative is to compute an estimate of $\widehat{E}_{k,R}^{(t)}$ at each site using consensus averaging, after which individual sites can compute SVD of $\widehat{E}_{k,R}^{(t)}$ locally. Despite its apparent simplicity, this approach will have significantly greater communication overhead compared to our proposed method.

[2] This can be accomplished, for example, through the use of (local) random number generators initialized with the same seed. Also, note that a key requirement in power method is that $\langle u_1, q^{init} \rangle \neq 0$, which is ensured with very high probability in the case of a random initialization.

eration. In order for this, we make use of the popular consensus averaging method [42]. To perform consensus averaging, we first design a doubly-stochastic weight matrix W that adheres to the topology of the underlying graph $\mathcal{G}$. In particular, we have that $w_{i,j} = 0$ whenever $(i,j) \notin \mathcal{E}$. We refer the reader to [42]–[44] for designing appropriate weight matrices in a distributed manner without relying on knowledge of the global network topology. In order to compute $\sum_i \widehat{M}_i^{(t)} \widehat{q}_i^{(t_p-1)}$ using consensus averaging, each site is initialized with $z_i^{(0)} = \widehat{M}_i^{(t)} \widehat{q}_i^{(t_p-1)}$. Next, let $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$ be the neighborhood of site i, define $Z^{(0)} = \begin{bmatrix} z_1^{(0)} & \dots & z_N^{(0)} \end{bmatrix}^\mathsf{T}$, and assume we are in t_c^{th} iteration of consensus averaging. Then consensus works by having each site carry out the following updates in each consensus iteration through communications with its neighbors: $z_i^{(t_c)} = \sum_{j \in \mathcal{N}_i} w_{i,j} z_j^{(t_c-1)}$. The dynamics of the overall system in this case evolve as $Z^{(t_c)} = W^{t_c} Z^{(0)}$. It then follows that $Z_{i,T}^{(t_c)} \xrightarrow{t_c} \mathbf{1}^\mathsf{T} Z^{(0)}/N$ [42], where $Z_{i,T}^{(t_c)}$ denotes the i^{th} row of $Z^{(t_c)}$ and $\mathbf{1} \in \mathbb{R}^N$ denotes a (column) vector of all ones. This in particular implies that each site achieves perfect consensus averaging as $t_c \to \infty$ and obtains $Z_{i,T}^{(\infty)^\mathsf{T}} = \frac{1}{N}\sum_{j=1}^N z_j^{(0)} = \frac{1}{N}\sum_{j=1}^N \widehat{M}_j^{(t)} \widehat{q}_j^{(t_p-1)}$.

But one can not perform infinite consensus iterations in practice within each iteration of the distributed power method. Instead, we assume a finite number of consensus iterations, denoted by T_c, in each power method iteration and make use of the modification of standard consensus averaging proposed in [38] to obtain $\widehat{v}_i^{(t_p)} = Z_{i,T}^{(T_c)^\mathsf{T}}/[W_1^{T_c}]_i$, where $W_1^{T_c}$ is the first column of W^{T_c} and $[\cdot]_i$ denotes the i^{th} entry of a vector. Note that this leads to an error $\epsilon_{i,c}^{(t_p)}$ within $\widehat{v}_i^{(t_p)}$ at each site for any finite T_c, i.e., $\widehat{v}_i^{(t_p)} = Z_{i,T}^{(T_c)^\mathsf{T}}/[W_1^{T_c}]_i = \sum_{j=1}^N \widehat{M}_j \widehat{q}_j^{(t_p-1)} + \epsilon_{i,c}^{(t_p)}$. After finishing consensus iterations, each site i in iteration t_p of power method normalizes this vector $\widehat{v}_i^{(t_p)}$ to get an estimate of the dominant eigenvector of $\widehat{M}^{(t)}$. Finally, we carry out enough iterations of the distributed power method at each site that the error between successive estimates of the eigenvector falls below a prescribed threshold.

We have now motivated and described the key components of our proposed algorithm and the full collaborative dictionary learning algorithm, termed *cloud K-SVD*, is detailed in Algorithm 1. Notice the initialization of cloud K-SVD differs from K-SVD in the sense that each site also generates a common (random) reference vector $d^{ref} \in \mathbb{R}^n$ and stores it locally. The purpose of d^{ref} is to ensure that the eigenvectors computed by different sites using the distributed power method all point in the same quadrant, rather than in antipodal quadrants (Step 18 in Algorithm 1). While this plays a role in analysis, it does not have an effect on the workings of cloud K-SVD. Notice also that we have not defined any stopping rules in Algorithm 1. One set of rules could be to run the algorithm for fixed dictionary learning iterations T_d, power method iterations T_p, and consensus iterations T_c. It is worth noting here that algorithms such as cloud K-SVD are often referred to as two time-scale algorithms in the literature. Nonetheless, cloud K-SVD with the stopping rules of finite (T_d, T_p, T_c) can be considered a *quasi* one time-scale algorithm. Accordingly, our analysis of cloud K-SVD assumes these stopping rules.

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 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 $\widehat{D}_i^{(t)} \leftarrow D^{init}, i = 1, \dots, N$.

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

Return: $\widehat{D}_i^{(t)}, i = 1, 2, \dots, N$.

Remark 1. A careful reading of Algorithm 1 reveals that normalization by $[W_1^{t_c}]_i$ in Step 15 is redundant due to the normalization in Step 16. We retain the current form of Step 15 however to facilitate the forthcoming analysis.

IV. Analysis of Cloud K-SVD

Since power method and consensus averaging in Algorithm 1 cannot be performed for an infinite number of iterations, in practice this results in residual errors in each iteration of the algorithm. It is therefore important to understand whether the dictionaries $\{\widehat{D}_i\}$ returned by cloud K-SVD approach the dictionary that could have been obtained by centralized K-SVD [7]. In order to address this question, we need to understand the behavior of major components of cloud K-SVD, 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

The first thing needed to quantify deviations of the cloud K-SVD dictionaries from the centralized K-SVD dictionary is algorithmic specification of the sparse coding steps in both algorithms. While the sparse coding steps as stated in (2) and (4) have combinatorial complexity, various low-complexity computational approaches can be used to solve these steps 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* [45]. Specifically, we assume sparse coding is carried out by solving

$$x_{i,s} = \arg\min_{x\in\mathbb{R}^K} \tfrac{1}{2}\|y_{i,s} - Dx\|_2^2 + \tau\|x\|_1 \tag{5}$$

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 [47]. Note that the lasso also has a dual, constrained form, given by

$$x_{i,s} = \arg\min_{x\in\mathbb{R}^K} \tfrac{1}{2}\|y_{i,s} - Dx\|_2^2 \quad \text{s.t.} \quad \|x\|_1 \leq \eta, \tag{6}$$

and (5) & (6) are identical for an appropriate $\eta_\tau = \eta(\tau)$ [48].

Remark 2. While extension of our analysis to other sparse coding methods such as *orthogonal matching pursuit* (OMP) [41] is beyond the scope of this work, such extensions would mainly rely on perturbation analyses of different sparse coding methods. In the case of OMP, for instance, such perturbation analysis is given in [46], which can then be leveraged to extend our lasso-based cloud K-SVD result to OMP-based result.

Our analysis in the following is also based on the assumption that cloud K-SVD and centralized K-SVD are identically initialized, i.e., $\widehat{D}_i^{(0)} = D^{(0)}, i = 1, \ldots, N$, where $D^{(t)}, t \geq 0$, in the following denotes the centralized K-SVD dictionary estimate in the t^{th} iteration. While both cloud K-SVD and centralized K-SVD start from the same initial estimates, 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 certain properties. Below, we present and motivate these properties.

[P1] Let $x_{i,s}^{(t)}$ denote the solution of the lasso (i.e., (5)) for $D = D^{(t-1)}$ and $\tau = \tau^{(t)}, t = 1, \ldots, T_d$. Then there exists some $C_1 > 0$ such that the following holds:

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

In our analysis in the following, we will also make use of the smallest regularization parameter among the collection $\left\{\tau^{(t)}\right\}_{t=1}^{T_d}$, defined as $\tau_{\min} = \min_t \tau^{(t)}$, and the largest dual parameter among the (dual) collection $\left\{\eta_\tau^{(t)} = \eta(\tau^{(t)})\right\}_{t=1}^{T_d}$, defined as $\eta_{\tau,\max} = \max_t \eta_\tau^{(t)}$.

[P2] Define $\Sigma_{T_0} = \left\{\mathcal{I} \subset \{1, \ldots, K\} : |\mathcal{I}| = T_0\right\}$. Then there exists some $C_2' > \frac{C_1^4\tau_{\min}^2}{1936}$ such that the following holds:

$$\min_{t=1,\ldots,T_d, \mathcal{I}\in\Sigma_{T_0}} \sigma_{T_0}\left(D_{|\mathcal{I}}^{(t-1)}\right) \geq \sqrt{C_2'},$$

where $\sigma_{T_0}(\cdot)$ denotes the T_0^{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 $\lambda_{1,k}^{(t)} > \lambda_{2,k}^{(t)} \geq \ldots \lambda_{n,k}^{(t)} \geq 0$ denote the eigenvalues of the centralized "reduced" matrix $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}, k \in \{1, \ldots, K\}$, in the t^{th} iteration, $t \in \{1, \ldots, T_d\}$. 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.

We now comment on the rationale behind these three properties. Properties P1 and P2 correspond to sufficient conditions for $x_{i,s}^{(t)}$ to be a unique solution of (5) [49] 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 $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$ in each dictionary learning iteration [26]. In particular, P3 is a statement about the worst-case spectral gap of $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$. 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 $\left\{E_k^{(t)}\right\}_{k=1}^K$ generated by the centralized K-SVD in each iteration. We define this parameter in the following for later use. Let $E_{i,k}^{(t)}, i = 1, \ldots, N$, denote part of the centralized error matrix $E_k^{(t)}$ associated with the data of the i^{th} site in the t^{th} iteration, i.e., $E_k^{(t)} = \begin{bmatrix} E_{1,k}^{(t)} & \cdots & E_{N,k}^{(t)}\end{bmatrix}, k = 1, \ldots, K, t = 1, \ldots, T_d$. Then

$$C_4 = \max\left\{1, \max_{t,i,k} \|E_{i,k}^{(t)}\|_2\right\}. \tag{7}$$

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, defined as

$$T_{mix} = \max_{i=1,\ldots,N} \inf_{t\in\mathbb{N}} \left\{t : \|\mathrm{e}_i^\mathsf{T}W^t - \tfrac{1}{N}\mathbf{1}^\mathsf{T}\|_2 \leq \frac{1}{2}\right\}. \tag{8}$$

Here, $\mathrm{e}_i \in \mathbb{R}^N$ denotes the i^{th} column of the identity matrix I_N. Note that the mixing time T_{mix} can be upper bounded in terms of inverse of the absolute spectral gap of W, defined as $1 - |\lambda_2(W)|$ with $\lambda_2(W)$ denoting the second largest (in modulus) eigenvalue of W [50]. As a general rule, better-connected networks can be made to have smaller mixing times compared to sparsely connected networks. We refer the reader to [51] and [50, Chap. 15] for further details on the relationship between T_{mix} and the underlying network topology.

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 $\alpha = \max_{t,k} \sum_{i=1}^{N} \|\widehat{E}_{i,k,R}^{(t)} \widehat{E}_{i,k,R}^{(t)^T}\|_2$, $\beta = \max_{t,t_p,k} \frac{1}{\left\|\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^T} q_{c,t,k}^{(t_p)}\right\|_2}$, $\gamma = \max_{t,k} \sqrt{\sum_{i=1}^{N} \|\widehat{E}_{i,k,R}^{(t)} \widehat{E}_{i,k,R}^{(t)^T}\|_F^2}$, $\nu = \max_{t,k} \frac{\widehat{\lambda}_{2,k}^{(t)}}{\widehat{\lambda}_{1,k}^{(t)}}$, $\widehat{\theta}_k^{(t)} \in [0, \pi/2]$ *as* $\widehat{\theta}_k^{(t)} = \arccos\left(\frac{\left|\left\langle u_{1,k}^{(t)}, q^{init}\right\rangle\right|}{\|u_{1,k}^{(t)}\|_2 \|q^{init}\|_2}\right)$, $\mu = \max\left\{1, \max_{k,t} \tan(\widehat{\theta}_k^{(t)})\right\}$, *and* $\zeta = K\sqrt{2S_{\max}}\left(\frac{6\sqrt{KT_0}}{\tau_{\min} C_2} + \eta_{\tau,\max}\right)$, *where* $S_{\max} = \max_i S_i$, $u_{1,k}^{(t)}$ *is the dominant eigenvector of* $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^T}$, $\widehat{\lambda}_{1,k}^{(t)}$ *and* $\widehat{\lambda}_{2,k}^{(t)}$ *are first and second largest eigenvalues of* $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^T}$ *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* $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^T}$. *Then, assuming* $\min_{t,k} \left|\langle u_{1,k}^{(t)}, q^{init}\rangle\right| > 0$, *and fixing any* $\epsilon \in \left(0, \min\left\{(10\alpha^2\beta^2)^{-1/3T_p}, (\frac{1-\nu}{4})^{1/3}\right\}\right)$ *and* $\delta_d \in \left(0, \min\left\{\frac{1}{\sqrt{2}}, \frac{C_1^2 \tau_{\min}}{44\sqrt{2K}}\right\}\right)$, *we have*

$$\max_{\substack{i=1,\ldots,N \\ k=1,\ldots,K}} \left\| \widehat{d}_{i,k}^{(T_d)} \widehat{d}_{i,k}^{(T_d)^T} - d_k^{(T_d)} d_k^{(T_d)^T} \right\|_2 \leq \delta_d \tag{9}$$

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

The proof of this theorem is given in Appendix C. 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. 9). 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 δ_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. Notice that the effect of network topology on the number of consensus iterations is captured through the dependence of T_c on the mixing time T_{mix}. 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 estimates of (centralized) $E_{1,R}^{(1)} E_{1,R}^{(1)^T}$ at individual sites, which then accumulate for $(k,t) \neq (1,1)$ to also cause errors in estimate $\widehat{E}_{k,R}^{(t)} \widehat{E}_{k,R}^{(t)^T}$ of the matrix $E_{k,R}^{(t)} E_{k,R}^{(t)^T}$ 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. Sec. III-B). The following result effectively helps us control the errors in cloud K-SVD dictionaries due to Steps 7–17 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* $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. *Suppose each* $M_i, i = 1, \ldots, N$, *is only available at the* i^{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 7–17 in Algorithm 1). Next, define* $\alpha_p = \sum_{i=1}^{N} \|M_i\|_2$, $\beta_p = \max_{t_p=1,\ldots,T_p} \frac{1}{\|M q_c^{(t_p)}\|_2}$, *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* $\epsilon \in \big(0, (10\alpha_p^2\beta_p^2)^{-1/3T_p}\big)$, *we have*

$$\max_{i=1,\ldots,N} \left\| u_1 u_1^T - \widehat{q}_i \widehat{q}_i^T \right\|_2 \leq \tan(\theta) \left|\frac{\lambda_2}{\lambda_1}\right|^{T_p} + 4\epsilon^{3T_p}, \tag{10}$$

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 10–14 in Algorithm 1) satisfies* $T_c = \Omega\big(T_p T_{mix} \log(2\alpha_p\beta_p\epsilon^{-1}) + T_{mix} \log(\alpha_p^{-1}\gamma_p\sqrt{N})\big)$. *Here,* θ *denotes the angle between* u_1 *and* q^{init}, *defined as* $\theta = \arccos(|\langle u_1, q^{init}\rangle|/(\|u_1\|_2 \|q^{init}\|_2))$.

The proof of this theorem is given in Appendix A. Theorem 2 states that $\widehat{q}_i \xrightarrow{T_p} \pm u_1$ geometrically 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, (10) 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 $\widehat{E}_{k,R}^{(t)} = \left[\widehat{E}_{1,k,R}^{(t)}, \cdots, \widehat{E}_{N,k,R}^{(t)}\right]$ and the reduced centralized error matrix $E_{k,R}^{(t)} = \left[E_{1,k,R}^{(t)}, \cdots, E_{N,k,R}^{(t)}\right]$ for $k \in \{1, \cdots, K\}$ and $t \in \{1, \cdots, T_d\}$. This result helps us control the error in step 5 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 Sec. III-A and Sec. III-B, respectively. Next, express $\Omega_k^{(t)} = diag\{\Omega_{1,k}^{(t)}, \cdots, \Omega_{N,k}^{(t)}\}$, where $\Omega_{i,k}^{(t)}$ corresponds to the data samples associated with the i^{th} site, and define $B_{i,k,R}^{(t)} = \widehat{E}_{i,k,R}^{(t)} - E_{i,k,R}^{(t)}$. Finally, let ζ, μ, ν, ϵ, and δ_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 $T_c = \Omega\big(T_pT_{mix}\log(2\alpha\beta\epsilon^{-1}) + T_{mix}\log(\alpha^{-1}\gamma\sqrt{N})\big)$ consensus iterations in cloud K-SVD and assume P1–P3 hold, we have for $i \in \{1, \ldots, N\}$, $t \in \{1, \cdots, T_d\}$, and $k \in \{1, \cdots, K\}$*

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

Proof of Theorem 3 along with the proofs of supporting lemmas is given in Appendix B. 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 ϵ as long as all of the assumptions of Theorem 1 are satisfied. The proof of Theorem 1, given in Appendix C, relies on these two aforementioned theorems. In particular, the proof of Theorem 1 shows that the assumption on ε in Theorem 3 is satisfied as long as we are performing power method iterations and consensus iterations as required by Theorem 1.

V. Numerical Experiments

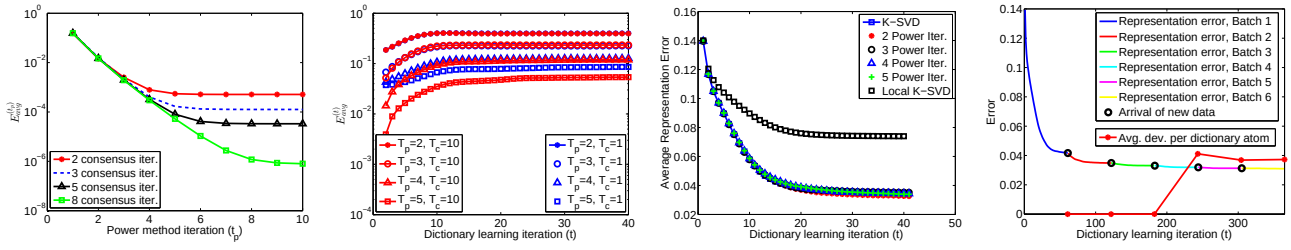
We present numerical results in this section for demonstrating the usefulness of cloud K-SVD and also validating some of our theoretical results. In the first set of experiments, synthetic data is used to demonstrate efficacy of cloud K-SVD for data representation. Furthermore, behavior of distributed power method (Steps 7–17 in Algorithm 1) as a function of the number of consensus iterations and deviations in cloud K-SVD dictionaries from centralized dictionary as a function of number of power method iterations are also shown with the help of simulations. In the second set of experiments, MNIST dataset is used to motivate an application of cloud K-SVD that can benefit from collaboration between distributed sites.

A. Experiments Using Synthetic Data

These experiments correspond to a total of $N = 100$ sites, with each site having $S_i = 500$ local samples in $\mathbb{R}^{20}$ (i.e., $n = 20$). Interconnections between the sites are randomly generated using an Erdős–Rényi graph with parameter $p = 0.5$. In order to generate synthetic data at individual sites, we first generate a dictionary with $K = 50$ atoms, $D \in \mathbb{R}^{20\times 50}$, with columns uniformly distributed on the unit sphere in $\mathbb{R}^{20}$. Next, we randomly select a 45-column subdictionary of D for each site and then generate samples for that site using a linear combination of $T_0 = 3$ randomly selected atoms of this subdictionary, followed by addition of white Gaussian noise with variance $\sigma^2 = 0.01$. All data samples in our experiments are also normalized to have unit ℓ_2 norms. Sparse coding in these experiments is performed using an implementation of OMP provided in [52]. Finally, in order to carry out distributed consensus averaging, we generate a doubly-stochastic weight matrix W according to the local-degree weights method described in [42, Sec. 4.2].

In our first set of experiments we illustrate the convergence behavior of the distributed power method component within cloud K-SVD (Steps 7–17 in Algorithm 1) as a function of the number of consensus iterations. The results of these experiments, which are reported in Fig. 2(a), correspond to five different values of the number of consensus iterations (3, 4, 5, 10, 15) within each iteration of the distributed power method. Specifically, let q denote the principal eigenvector of the matrix $\sum_{i=1}^{N}\widehat{M}_i$ in Algorithm 1 (Step 6) computed using Matlab (ver. 2014a) and $\widehat{q}_i^{(t_p)}$ denote an estimate of q obtained at site i after the t_p^{th} iteration of the distributed power method. Then Fig. 2(a) plots $E_{\text{eig}}^{(t_p)}$, which is the average of $\|qq^{\mathsf{T}} - \widehat{q}_i^{(t_p)}\widehat{q}_i^{(t_p)\mathsf{T}}\|_2$ over all sites $i \in \{1, \ldots, N\}$, dictionary update steps $k \in \{1, \ldots, K\}$, dictionary learning iterations T_d, and 100 Monte-Carlo trials, as a function of the number of distributed power method iterations t_p. It can be seen from this figure that the distributed power method of Algorithm 1 hits an *error floor* with increasing number of distributed power method iterations, where the floor is fundamentally determined by the number of consensus iterations within each power method iteration, as predicted by Theorem 2.

Using the same setup our second set of experiments demonstrate the effectiveness of collaboratively learning a dictionary using cloud K-SVD, as opposed to each site learning a *local dictionary* from its local data using the canonical K-SVD algorithm (referred to as *local K-SVD* in the following). Moreover, these experiments also demonstrate the variations in cloud K-SVD results when we change the number of power method iterations (T_p) and consensus iterations (T_c). In Fig. 2(c), we plot average representation error, defined as $\frac{1}{nS}\sum_{i=1}^{N}\sum_{j=1}^{S_i}\|y_{i,j} - Dx_{i,j}\|_2$, as a function of the number of dictionary learning iterations for three dictionary learning methods, namely, centralized (canonical) K-SVD, cloud K-SVD, and local K-SVD. It can be seen from this figure, which corresponds to an average of 100 Monte-Carlo trials, that cloud K-SVD and centralized K-SVD have similar performance and both of them perform better than local K-SVD. In particular, the local K-SVD error is ≈ 0.06 after 40 iterations, while



(a) Distributed Power Method (b) Cloud K-SVD Error (c) Average Representation Error (d) Online K-SVD

Fig. 2. Performance of cloud K-SVD on synthetic data. (a) Average error in eigenvector estimates of distributed power method. (b) Average error in dictionary atoms returned by cloud K-SVD. (c) Average representation error of cloud K-SVD. (d) Average representation error and average deviation per dictionary atom (from centralized dictionary learned in a full-batch setting) of K-SVD in an online setting as a function of dictionary learning iterations.

it is ≈ 0.03 for cloud K-SVD and centralized K-SVD. Notice that changes in the number of power method iterations induce relatively minor changes in the representation error of cloud K-SVD. Next, Fig. 2(b) highlights the average error in dictionary atoms learned using cloud K-SVD as compared to centralized K-SVD. For this experiment, number of consensus iterations are either $T_c = 1$ or $T_c = 10$, and for each of these values, the number of power method iterations used are $T_p = 2, 3, 4, 5$. These experiments show the effect of changing T_p and T_c on the error in collaborative dictionaries. This error is averaged over all dictionary atoms and sites in each iteration for 100 Monte-Carlo trials, defined as $E^{(t)}_{\text{average}} = \frac{1}{NK}\sum_{k=1}^{K}\sum_{i=1}^{N}\|d_k^{(t)}d_k^{(t)^{\mathsf{T}}} - \widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^{\mathsf{T}}}\|_2$. Results in Fig. 2(b) show that this error in dictionary atoms increases sharply at the start, but it stabilizes after some iterations. Important point to note here is that as we increase the number of power method iterations and consensus iterations we get smaller average error in dictionary atoms as predicted by our analysis.

Next, we discuss the usage of cloud K-SVD in online settings. Since it has already been demonstrated that cloud K-SVD achieves performance similar to that of K-SVD, we focus here on the representation error of centralized K-SVD in online settings. The setup corresponds to a mini-batch of 500 training samples being periodically generated at each site and the assumption that each site has a buffer limit of 1000 samples. Thus only samples from the last two periods can be used for dictionary learning. After arrival of each new mini-batch of training samples, we use the dictionary learned in the last period to warm-start (centralized) K-SVD and carry out 60 dictionary learning iterations. Fig. 2(d) shows the representation error of the learned dictionary in this case, along with the deviation per dictionary atom when compared to a dictionary learned using full-batch centralized K-SVD. These results are plotted as a function of dictionary learning iterations for six periods, where the ending of a period is marked by a circle. The representation error curve in this figure shows that K-SVD takes more time to converge, but it (and thus cloud K-SVD) is a viable option for online settings. Similarly, the deviation curve shows that while the dictionary error initially increases with the arrival of more data, it stabilizes afterward. Note that further improvements in these results can be obtained by using methods like [53] for active sample selection.

Finally, we perform experiments to report actual values of the parameters C_1–C_4. To this end, we generate samples belonging to $\mathbb{R}^{17}$, where each sample is a linear combination of $T_0 = 3$ atoms of a dictionary $D \in \mathbb{R}^{17\times 40}$. We perform sparse coding in these experiments using the lasso package in Matlab 2014a, while we perform dictionary learning using K-SVD. Average values obtained for parameters C_1–C_4 over 100 Monte-Carlo trials in this case are 0.0586, 0.1633, 4.544, and 1.5947, respectively. Using cloud K-SVD, average values of μ and ν are 9000 and 0.3242, respectively. Based on these values, we get $T_p \approx 16,000$. This suggests that the constants in our bounds are rather loose, and our analysis should mainly be used to provide scaling guidelines.

B. Classification of MNIST Images

For evaluation of cloud K-SVD on real dataset, we perform classification of digits $\{0, 3, 5, 8, 9\}$ from MNIST dataset [54]. For each digit 6000 samples are used, where 5000 samples are used for training purposes and remaining 1000 for testing purposes. The data are five-times randomly split into training and test samples. For cloud K-SVD, Erdős–Rényi graph with parameter $p = 0.5$ is used to generate a network with 10 sites and data is equally distributed among them. Before performing dictionary learning, data is down sampled from $\mathbb{R}^{784}$ to $\mathbb{R}^{256}$. After downsampling, a separate dictionary is learned for each digit using centralized K-SVD, cloud K-SVD, and K-SVD using only local data. Each dictionary has dimensions $\mathbb{R}^{256\times 400}$, i.e., $K = 400$, and sparsity level of $T_0 = 10$ is used. Minimum residue based rule [55, Sec.II-A] is used for classification, more details on which are given in the following paragraph.

Let $\{D_c\}_{c=1}^{5}$ be the set of dictionaries for 5 classes and let $D = \begin{bmatrix} D_1 & D_2 & D_3 & D_4 & D_5 \end{bmatrix}$ be the complete dictionary. For any test sample y_s, we perform sparse coding using dictionary D with sparsity constraint of $T_0 = 10$ to get coefficients $x_s \in \mathbb{R}^{2000}$. Then we partition x_s into five segment $\{x_{s,c}\}_{c=1}^{5}$, where $x_{s,c}$ are the coefficients corresponding to dictionary D_c of class c. Next we define residue for class c as $r_c = \|y_s - D_c x_{s,c}\|_2$. Finally, the detected class is given by $c^* = \arg\min_c r_c$. Performance of each method (centralized K-SVD, cloud K-SVD, and local K-SVD) is measured in terms of average detection rate on the test samples, which is defined as $R_c = \frac{\text{Number of samples in class } c \text{ detected correctly}}{\text{Total number of samples of class } c}$. Results of this experiment are given in Fig. 3. We see that centralized and cloud K-SVD have comparable performance. But in the case of local K-SVD where we only use the local data for learning representations, classification rate deteriorates considerably. The bars in local K-SVD show the highest and lowest

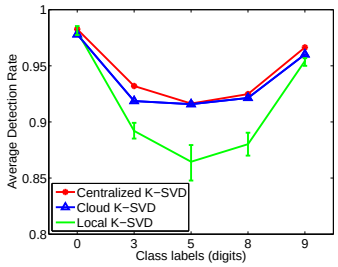


Fig. 3. Average detection rate for five classes of MNIST dataset using centralized K-SVD, cloud K-SVD, and local K-SVD.

detection rates achieved among the 10 sites, which highlights the variation in effectiveness of models learned across different sites when using only the local data.

VI. Conclusion

In this paper, we have proposed a new dictionary learning algorithm, termed cloud K-SVD, that facilitates collaborative learning of a dictionary that best approximates massive data distributed across geographical regions. Mathematical analysis of proposed method is also provided, which under certain assumptions shows that if we perform enough number of power method and consensus iterations then the proposed algorithm converges to the centralized K-SVD solution. Furthermore, the efficacy of the proposed algorithm is demonstrated through extensive simulations on synthetic and real data.

Appendix A
Proof of Theorem 2

The proof of this theorem relies on a lemma that guarantees that if the estimates obtained at different sites using the distributed power method are close to the estimate obtained using the centralized power method at the start of a power method iteration then the distributed estimates remain close to the centralized estimate at the end of that iteration. To prove such a lemma, we first need a result from the literature that characterizes the convergence behavior of *vector* consensus averaging as a function of the number of consensus iterations.

Proposition 1. *[38, Theorem 5] Consider the $n\times 1$ vector sum $z=\sum_{i=1}^N z_i^{(0)}$ and suppose each vector $z_i^{(0)}, i=1,\dots,N$, is only available at the i^{th} site in our network. Let b be a vector whose entries are the sum of absolute values of the initial vectors $z_i^{(0)}$ (i.e., j^{th} entry of b is $b_j=\sum_{i=1}^N |z_{i,j}^{(0)}|$) and $z_i^{(t_c)}$ be the $n\times 1$ vector obtained at the i^{th} site after t_c consensus iterations. Then, fixing any $\delta>0$, we have that $\left\| \frac{z_i^{(t_c)}}{[W_1^{t_c}]_i} - z\right\|_2 \le \delta\|b\|_2\ \forall i$ as long as the number of consensus iterations satisfies $t_c=\Omega(T_{mix}\log\delta^{-1})$.*

We use Proposition 1 to state and prove the desired lemma.

Lemma 1. *Suppose we are at the start of $(t_p+1)\le T_p$ power method iteration. Let q_c and $q_{i,d}$ denote the outputs of centralized power method and distributed power method at i^{th} site after t_p iterations, respectively. Similarly, let q_c' and $q_{i,d}'$ denote the outputs of centralized power method and distributed power method at i^{th} site after t_p+1 iterations, respectively. Next, fix an $\epsilon\in(0,1)$, define $\delta=\frac{\alpha_p}{\gamma_p\sqrt{N}}\left(\frac{\epsilon}{2\alpha_p\beta_p}\right)^{3T_p}$, and assume that $\forall i,\ \|q_c-q_{i,d}\|_2+\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\le\frac{1}{2\alpha_p\beta_p^2(2\alpha_p+\delta\gamma_p\sqrt{N})}$. Then, assuming $\Omega(T_{mix}\log\delta^{-1})$ consensus iterations, we have that $\forall i,\ \|q_c'-q_{i,d}'\|_2\le(2\alpha_p\beta_p)^3\left(\max_{i=1,\dots,N}\|q_c-q_{i,d}\|_2+\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\right)$.*

Proof: Define $v=Mq_c$ and $\widehat{v}=\sum_{i=1}^N M_i q_{i,d}$. Next, fix any $i\in\{1,\dots,N\}$ and let $\widehat{v}_i$ be the vector obtained at the i^{th} site in Step 15 of Algorithm 1 during the (t_p+1) iteration of distributed power method. Notice that $\widehat{v}_i$ can be expressed as $\widehat{v}_i=\widehat{v}+\epsilon_{i,c}$, where $\epsilon_{i,c}$ denotes the error introduced in $\widehat{v}$ at the i^{th} site due to finite number of consensus iterations. Next, define $r=\|v\|_2$ and $\widehat{r}_i=\|\widehat{v}_i\|_2$ and notice that $q_c'-q_{i,d}'=v(r^{-1}-\widehat{r}_i^{-1})+(v-\widehat{v}_i)\widehat{r}_i^{-1}$. It therefore follows from the triangle inequality that

$$\|q_c'-q_{i,d}'\|_2\le\|v\|_2|r^{-1}-\widehat{r}_i^{-1}|+\|v-\widehat{v}_i\|_2\widehat{r}_i^{-1}. \tag{11}$$

We now need to bound $\|v\|_2$, $|r^{-1}-\widehat{r}_i^{-1}|$, $\|v-\widehat{v}_i\|_2$, and $\widehat{r}_i^{-1}$. To this end, notice that $v-\widehat{v}_i=\left[\sum_{i=1}^N M_i(q_c-q_{i,d})\right]-\epsilon_{i,c}$. It also follows from Proposition 1 and some manipulations that $\|\epsilon_{i,c}\|_2\le\delta\gamma_p\sqrt{N}$. We therefore obtain

$$\|v-\widehat{v}_i\|_2\le\sum_{i=1}^N\|M_i\|_2\|q_c-q_{i,d}\|_2+\delta\gamma\sqrt{N}. \tag{12}$$

Next, notice $|r^{-1}-\widehat{r}_i^{-1}|=|r-\widehat{r}_i|r^{-1}\widehat{r}_i^{-1}$ and further it can be shown that $|r-\widehat{r}_i|\le r^{-1}|\widehat{r}_i^2-r^2|$. Now, $|\widehat{r}_i^2-r^2|=|\widehat{v}_i^{\mathsf{T}}\widehat{v}_i-v^{\mathsf{T}}v|\le\|\widehat{v}_i-v\|_2(\|\widehat{v}_i\|_2+\|v\|_2)$. Since $\widehat{v}_i=\widehat{v}+\epsilon_{i,c}$, it can also be shown that $\|\widehat{v}_i\|_2\le\alpha_p+\delta\gamma_p\sqrt{N}$. In addition, we have $\|v\|_2\le\alpha_p$. Combining these facts with (12), we get

$$\begin{aligned}&|\widehat{r}_i^2-r^2|\\&\le(2\alpha_p+\delta\gamma_p\sqrt{N})\left(\sum_{i=1}^N\|M_i\|_2\|q_c-q_{i,d}\|_2+\delta\gamma_p\sqrt{N}\right),\\&\le(2\alpha_p+\delta\gamma_p\sqrt{N})\left(\alpha_p\max_i\|q_c-q_{i,d}\|_2+\delta\gamma_p\sqrt{N}\right).\end{aligned}\tag{13}$$

We can now use this inequality to obtain $|r^{-1}-\widehat{r}_i^{-1}|\le\widehat{r}_i^{-1}\beta_p^2(2\alpha_p+\delta\gamma_p\sqrt{N})(\alpha_p\max_i\|q_c-q_{i,d}\|_2+\delta\gamma_p\sqrt{N})$.

The only remaining quantity we need to bound is $\widehat{r}_i^{-1}$. To this end, notice that $|r-\widehat{r}_i|\ge(r^{-1})^{-1}-(\widehat{r}_i^{-1})^{-1}$. Since $|r-\widehat{r}_i|\le r^{-1}|\widehat{r}_i^2-r^2|$, we obtain from (13) that

$$\begin{aligned}(r^{-1})^{-1}-(\widehat{r}_i^{-1})^{-1}\le&\alpha_p r^{-1}(2\alpha_p+\delta\gamma_p\sqrt{N})\\&\left(\max_i\|q_c-q_{i,d}\|_2+\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\right).\end{aligned}$$

It then follows from the lemma's assumptions along with some algebraic manipulations that $\widehat{r}_i^{-1}\le 2\beta_p$. Finally, plugging the bounds on $\widehat{r}_i^{-1}$, $|r^{-1}-\widehat{r}_i^{-1}|$, $\|v\|_2$, and $\|v-\widehat{v}_i\|_2$ in (11), we obtain

$$\begin{aligned}&\|q_c'-q_{i,d}'\|_2\\&\le2\alpha_p\beta_p^3\left(\alpha_p\max_i\|q_c-q_{i,d}\|_2+\delta\gamma_p\sqrt{N}\right)\\&(2\alpha_p+\delta\gamma_p\sqrt{N})+2\beta_p\left(\alpha_p\max_i\|q_c-q_{i,d}\|_2+\delta\gamma_p\sqrt{N}\right)\\&=\left(4\alpha_p^3\beta_p^3+2\alpha_p^3\beta_p^3\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}+2\alpha_p\beta_p\right)\\&\quad\left(\max_i\|q_c-q_{i,d}\|_2+\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\right).\end{aligned}$$

Finally, $\frac{\delta\gamma_p\sqrt{N}}{\alpha_p} \leq \left(\frac{\epsilon}{2}\right)^{3T_p} < 1$ since (i) $\delta = \frac{\alpha_p}{\gamma_p\sqrt{N}}\left(\frac{\epsilon}{2\alpha_p\beta_p}\right)^{3T_p}$, (ii) $\epsilon < 1$, and (iii) $\alpha_p r^{-1} \geq 1$, which implies $\alpha_p\beta_p \geq 1$. Plugging this into the above expression and noting that $\alpha_p\beta_p \leq \alpha_p^3\beta_p^3$, we obtain the claimed result. ∎

Lemma 1 provides an understanding of the error accumulation in the distributed power method due to finite number of consensus iterations in each power method iteration. And while the factor of $(2\alpha_p\beta_p)^3$ in the lemma statement might seem discouraging, the fact that the distributed power method starts with a zero error helps keep the total error in control. We now formally argue this in the proof of Theorem 2 below.

Proof of Theorem 2: We begin by defining q_{c} as the estimate of u_1 obtained using T_p iterations of the centralized power method that is initialized with the same q^{init} as the distributed power method. Next, fix an $i \in \{1, \ldots, N\}$ and notice that

$$\left\|u_1u_1^\mathsf{T} - \widehat{q}_i\widehat{q}_i^\mathsf{T}\right\|_2 \leq \|u_1u_1^\mathsf{T} - q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T}\|_2 + \|q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T} - \widehat{q}_i\widehat{q}_i^\mathsf{T}\|_2. \quad (14)$$

The convergence rate of the centralized power method is well studied and can be expressed as [26]

$$\|u_1u_1^\mathsf{T} - q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T}\|_2 \leq \tan(\theta)\left|\frac{\lambda_2}{\lambda_1}\right|^{T_p}. \quad (15)$$

In order to bound $\|q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T} - \widehat{q}_i\widehat{q}_i^\mathsf{T}\|_2$, we make use of Lemma 1. To invoke this lemma, we first need to show that the main assumption of the lemma holds for all iterations $t_p \leq (T_p - 1)$. We start with $t_p = 0$ for this purpose and note that $q_{\mathsf{c}}^{(0)} = \widehat{q}_i^{(0)} = q^{init}$, which trivially implies $\|q_{\mathsf{c}}^{(0)} - \widehat{q}_i^{(0)}\|_2 + \frac{\delta\gamma_p\sqrt{N}}{\alpha_p} \leq (\frac{\epsilon}{2})^{3T_p}$, where δ is as defined in Lemma 1. Further, under the assumptions of the theorem, it can be shown through elementary algebra that $\left(\frac{\epsilon}{2}\right)^{3T_p} \leq \frac{1}{2\alpha_p\beta_p^2(2\alpha_p + \delta\gamma_p\sqrt{N})}$. We now invoke mathematical induction and claim that the main assumption of Lemma 1 is satisfied for all $t_p \leq m < T_p$. Then we obtain from a recursive application of the statement of the lemma that for $t_p = (m+1)$, we have

$$\begin{aligned}
&\|q_{\mathsf{c}}^{(m+1)} - \widehat{q}_i^{(m+1)}\|_2 + \frac{\delta\gamma_p\sqrt{N}}{\alpha_p} \\
&\leq \frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\sum_{i=0}^{m}(2\alpha_p\beta_p)^{3i} \overset{(a)}{\leq} 2\cdot\frac{\delta\gamma_p\sqrt{N}}{\alpha_p}(2\alpha_p\beta_p)^{3m} \\
&= 2\cdot\epsilon^{3T_p}\frac{(2\alpha_p\beta_p)^{3m}}{(2\alpha_p\beta_p)^{3T_p}} \overset{(b)}{\leq} \frac{1}{2\alpha_p\beta_p^2(2\alpha_p + \delta\gamma_p\sqrt{N})}, \quad (16)
\end{aligned}$$

where (a) follows from the geometric sum and the fact that $(2\alpha_p\beta_p)^3 > 2$, while (b) follows from the theorem assumptions and the fact that $m < T_p$. We have now proved that the main assumption of Lemma 1 holds for all $t_p \leq (T_p - 1)$. In order to compute $\|q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T} - \widehat{q}_i\widehat{q}_i^\mathsf{T}\|_2$, therefore, we can recursively apply the result of this lemma up to the T_p^{th} iteration to obtain

$$\|q_{\mathsf{c}} - \widehat{q}_i\|_2 \leq \frac{\delta\gamma_p\sqrt{N}}{\alpha_p}\sum_{i=0}^{T_p}(2\alpha_p\beta_p)^{3i} \overset{(c)}{\leq} 2\epsilon^{3T_p}, \quad (17)$$

where (c) follows from the same arguments as in (16). The proof of the theorem now follows by noting the fact that $\|q_{\mathsf{c}}q_{\mathsf{c}}^\mathsf{T} - \widehat{q}_i\widehat{q}_i^\mathsf{T}\|_2 \leq (\|q_{\mathsf{c}}\|_2 + \|\widehat{q}_i\|_2)\|q_{\mathsf{c}} - \widehat{q}_i\|_2 \leq 4\epsilon^{3T_p}$. ∎

APPENDIX B
PROOF OF THEOREM 3

Notice from Algorithm 1 that sparse coding is always performed before update of the first dictionary atom. However, we do not perform sparse coding before updating any other dictionary atom. Due to this distinction, we answer how error is accumulated in matrix $E_{i,k,R}^{(t)}$ for first dictionary atom differently than for any other dictionary atom. In the following, we first provide an overview of how to bound $\|B_{i,k+1,R}^{(t)}\|_2$ when we know a bound on $\|B_{i,k,R}^{(t)}\|_2$. Then we will talk about bounding $\|B_{i,1,R}^{(t+1)}\|_2$ when we know bounds on $\{\|B_{i,j,R}^{(t)}\|_2\}_{j=1}^K$.

Recall from Step. 5 in Algorithm 1 that $\widehat{E}_{i,k,R}^{(t)} = Y_i\widetilde{\Omega}_{i,k}^{(t)} - \sum_{j=1}^{k-1}\widehat{d}_{i,j}^{(t)}\widehat{x}_{i,j,T}^{(t)}\widetilde{\Omega}_{i,k}^{(t)} - \sum_{j=k+1}^{K}\widehat{d}_{i,j}^{(t-1)}\widetilde{x}_{i,j,T}^{(t)}\widetilde{\Omega}_{i,k}^{(t)}$. Now, if one assumes that $\widetilde{\Omega}_k^{(t)} = \Omega_k^{(t)}$, which we will argue is true, then the error in $E_{i,k,R}^{(t)}$ is due to errors in $\{x_{i,j,T,R}^{(t)}\}_{j=1}^K$ and $\{d_j^{(t)}\}_{j=1}^K$. Infact, we will show that $\|B_{i,k+1,R}^{(t)}\|_2$ can be bounded by knowing bounds on errors in $\widehat{d}_{i,k}^{(t)}$ and $x_{i,k,T,R}^{(t)}$ only. Next, recall from Step. 19 in Algorithm 1 that $\widehat{x}_{i,k,R}^{(t)} = \widehat{d}_{i,k}^{(t)}\widehat{E}_{i,k,R}^{(t)^\mathsf{T}}$, which means we only need to know a bound on $d_k^{(t)}$ to bound $\|B_{i,k+1,R}^{(t)}\|_2$. Another challenge for us will be to bound error in $d_k^{(t)}$ from a given bound on $\|B_{i,k,R}^{(t)}\|_2$. We will accomplish this by noting that there are two sources of error in $\widehat{d}_k^{(t)}$. The first source is the difference in eigenvectors of $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}}$ and $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$. We will bound this difference using Proposition 3 in Appendix D. In order to use this proposition, we will need a bound on $\|\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}} - E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}\|_F$, which we will also prove using a given bound on $\|B_{i,k,R}^{(t)}\|_2$ (Lemma 2). The second source of error in $\widehat{d}_k^{(t)}$ 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 ε. Combining these two sources of error, we will first bound the error in $\widehat{d}_k^{(t)}$ (Lemma 3), and then using this we will finally bound $\|B_{i,k+1,R}^{(t)}\|_2$ (Lemma 5).

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 the bounds on $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$ derived earlier (Lemma 3). Next, using error bounds on $\{\widehat{d}_{i,j}^{(t)}\}_{j=1}^K$, we can use Proposition 2 in Appendix D 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 will bound $\|B_{i,1,R}^{(t+1)}\|_2$ (Lemma 4). This will be followed by the remaining proof of Theorem 3.

Our first result in support of Theorem 3 shows that the assumption of Proposition 3 in Appendix D is satisfied under certain conditions, which will make it possible for us to bound

the difference in the principal eigenvector of $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$ and $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}}$.

Lemma 2. *Let $\Omega_{i,k}^{(t)}$, $\widetilde{\Omega}_{i,k}^{(t)}$, ε and ζ be as defined in Theorem 3. Fix δ_d as in Theorem 1, and suppose (i) P1–P3 are satisfied, (ii) $\Omega_{i,k}^{(t)} = \widetilde{\Omega}_{i,k}^{(t)}$, and (iii) $\varepsilon \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)}}$. Then $\forall i \in \{1,\ldots,N\}$ and for any $t \in \{1,\cdots,T_d\}$ and $k \in \{1,\cdots,K\}$, if*

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

then $\Delta M_k^{(t)} = E_{k,R}^{(t)}E_{k,R}^{(t)^T} - \widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^T}$ is bounded as $\|\Delta M_k^{(t)}\|_F \leq \frac{1}{5C_3}$.

Proof: Since our starting dictionaries are same, therefore, for $(t,k) = (1,1)$ we have $E_{1,R}^{(1)} = \widehat{E}_{1,R}^{(1)}$, which means $\Delta M_k = 0$. Hence, claim is true for $(t,k) = (1,1)$. In the following, proof is provided for the claim for case $(t,k) \neq 1$.

Substituting $B_{i,k,R}^{(t)}$ in the definition of $\Delta M_k^{(t)}$, we get

$$\Delta M_k^{(t)} = \sum_{i=1}^{N} E_{i,k,R}^{(t)}B_{i,k,R}^{(t)^\mathsf{T}} + B_{i,k,R}^{(t)}E_{i,k,R}^{(t)^\mathsf{T}} + B_{i,k,R}^{(t)}B_{i,k,R}^{(t)^\mathsf{T}}.$$

Simple algebraic manipulations, along with submultiplicativity of matrix 2-norm, result in

$$\begin{aligned}\|\Delta M_k^{(t)}\|_2 &\leq 2\sum_{i=1}^{N}\left(\|E_{i,k,R}^{(t)}\|_2\|B_{i,k,R}^{(t)}\|_2 + \|B_{i,k,R}^{(t)}\|_2^2\right) \\ &\leq 2N\max_i\left(C_4\|B_{i,k,R}^{(t)}\|_2 + \|B_{i,k,R}^{(t)}\|_2^2\right), \quad (18)\end{aligned}$$

where the last inequality is due to (7). Now, using the assumptions on bound of $\|B_{i,k,R}^{(t)}\|_2$ and ε, we get

$$\begin{aligned}&\|\Delta M_k^{(t)}\|_2 \\ &\leq 2N\varepsilon(1+\zeta)^{t-1}\Big(C_4^2(8C_3NC_4^2+5)^{(t-1)K+k-2} \\ &\quad +\varepsilon(1+\zeta)^{t-1}C_4^2(8C_3NC_4^2+5)^{2(t-1)K+2k-4}\Big) \\ &\leq 2N\varepsilon(1+\zeta)^{t-1}\Big(C_4^2(8C_3NC_4^2+5)^{(t-1)K+k-2} \\ &\quad +\frac{1}{8N\sqrt{n}C_3}\frac{(1+\zeta)^{t-1}C_4^2(8C_3NC_4^2+5)^{2(t-1)K+2k-4}\delta_d}{(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)K}}\Big) \\ &\overset{(a)}{\leq} 2N\varepsilon(1+\zeta)^{t-1}\left(C_4^2(8C_3NC_4^2+5)^{(t-1)K+k-2} + \frac{1}{8N\sqrt{n}C_3}\right) \\ &\leq 4N\varepsilon(1+\zeta)^{t-1}C_4^2(8C_3NC_4^2+5)^{(t-1)K+k-2},\end{aligned}$$

where (a) is true because $\frac{(1+\zeta)^{t-1}C_4^2(8C_3NC_4^2+5)^{2(t-1)K+2k-4}\delta_d}{(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)}} \leq 1$. Finally, using once again the assumption on ε, performing algebraic manipulations and using the fact that $\delta_d \leq 1$, we get

$$\begin{aligned}\|\Delta M_k^{(t)}\|_2 &\leq \frac{(8C_3NC_4^2+5)^{(t-1)K+k-2}}{2\sqrt{n}C_3(8C_3NC_4^2+5)^{2(T_dK-2)}} \\ &\leq \frac{1}{\sqrt{n}(8C_3NC_4^2+5)^{(T_dK-2)}} \leq \frac{1}{\sqrt{n}(5C_3)}.\end{aligned}$$

Now using the fact that $\text{rank}(\Delta M_k^{(t)}) \leq n$, we get $\|\Delta M_k^{(t)}\|_F \leq \sqrt{\text{rank}(\Delta M_k^{(t)})}\|\Delta M_k^{(t)}\|_2 \leq \sqrt{n}\|\Delta M_k^{(t)}\|_2 \leq \frac{1}{5C_3}$. ∎

We are now ready to prove that if we know a bound on $\|B_{i,k,R}^{(t)}\|_2$ then we can bound the error in dictionary atom $\widehat{d}_{i,k}^{(t)}$. This result is given in the following lemma.

Lemma 3. *Let $\Omega_{i,k}^{(t)}$, $\widetilde{\Omega}_{i,k}^{(t)}$, ε and ζ be as defined in Theorem 3, also perform T_c consensus iterations as given in Theorem 3. Now fix δ_d as in Theorem 1, and suppose (i) P1–P3 are satisfied, (ii) $\Omega_{i,k}^{(t)} = \widetilde{\Omega}_{i,k}^{(t)}$, and (iii) $\varepsilon \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)}}$. Then for all $i \in \{1,\ldots,N\}$ and for any $t \in \{1,2,\cdots,T_d\}$ and $k \in \{1,2,\cdots,K\}$ if we know*

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

then, $\|\widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^T} - d_k^{(t)}d_k^{(t)^T}\|_2 \leq \varepsilon(1+\zeta)^{t-1}(8C_3NC_4^2+5)^{(t-1)K+k-1}$.

Proof: To prove this lemma we first need to decompose error in dictionary atom into two different components i.e., error in principal eigenvector due to perturbation in $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$ and error due to distributed power method. Let $d_k^{(t)}$ be the updated k^{th} atom of centralized dictionary at iteration t, which is the principal eigenvector of $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$. In cloud K-SVD, $\widehat{d}_{i,k}^{(t)}$ corresponds to the principal eigenvector estimate of $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}}$ obtained at the i^{th} site. Let us denote the true principal eigenvector of $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}}$ by $\widetilde{d}_k^{(t)}$ and let $\widehat{d}_{i,k}^{(t)}$ be the eigenvector of $\widehat{E}_{k,R}^{(t)}\widehat{E}_{k,R}^{(t)^\mathsf{T}}$ computed using distributed power method at the i^{th} site. Using this notation, notice that $\|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}}\|_2 \leq \|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widetilde{d}_k^{(t)}\widetilde{d}_k^{(t)^\mathsf{T}}\|_2 + \|\widetilde{d}_k^{(t)}\widetilde{d}_k^{(t)^\mathsf{T}} - \widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}}\|_2$, where the first term is due to perturbation in $E_{k,R}^{(t)}E_{k,R}^{(t)^\mathsf{T}}$ and the second term is due to imperfect power method and consensus iterations. We can now use Theorem 2 to obtain

$$\begin{aligned}&\|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}}\|_2 \\ &\leq \|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widetilde{d}_k^{(t)}\widetilde{d}_k^{(t)^\mathsf{T}}\|_2 + \tan(\widehat{\theta}_k^{(t)})\left(\frac{\widehat{\lambda}_{2,k}^{(t)}}{\widehat{\lambda}_{1,k}^{(t)}}\right)^{T_p} + 4\epsilon^{3T_p} \\ &\overset{(a)}{\leq} \|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widetilde{d}_k^{(t)}\widetilde{d}_k^{(t)^\mathsf{T}}\|_2 + \mu\nu^{T_p} + 4\epsilon^{3T_p} \\ &\overset{(b)}{=} \|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widetilde{d}_k^{(t)}\widetilde{d}_k^{(t)^\mathsf{T}}\|_2 + \varepsilon,\end{aligned}$$

where (a) is due to definition of parameters μ and ν in Theorem 1, and (b) is due to definition of ε in Theorem 3.

Next, for symmetric matrices $M_k^{(t)} = \sum_i E_{i,k,R}^{(t)}E_{i,k,R}^{(t)^\mathsf{T}}$ and $\widehat{M}_k^{(t)} = \sum_i \widehat{E}_{i,k,R}^{(t)}\widehat{E}_{i,k,R}^{(t)^\mathsf{T}}$ such that $\widehat{M}_k^{(t)} = M_k^{(t)} + \Delta M_k^{(t)}$, we can use Lemma 2 and Proposition 3 to find a bound on deviation in principal eigenvector of $M_k^{(t)}$ due to perturbation $\Delta M_k^{(t)}$. Since we have from Lemma 2 that $\|\Delta M_k^{(t)}\|_F \leq \frac{1}{5C_3}$, it follows from Proposition 3 that

$$\begin{aligned}\|d_k^{(t)}d_k^{(t)^\mathsf{T}} - \widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}}\|_2 &\leq 4C_3\|\Delta M_k^{(t)}\|_2 + \varepsilon \\ &\leq 8C_3N\max_i\left(C_4\|B_{i,k,R}^{(t)}\|_2 + \|B_{i,k,R}^{(t)}\|_2^2\right) + \varepsilon, \quad (19)\end{aligned}$$

where the last inequality is due to (18). Now using the bound on $\|B_{i,k,R}^{(t)}\|_2$ in the lemma statement, it can be shown using some algebraic manipulations that

$$\begin{aligned}&\|\widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}} - d_k^{(t)}d_k^{(t)^\mathsf{T}}\|_2 \\ &\quad\le \varepsilon(1+\zeta)^{t-1}C_4\Big(8C_3NC_4^2(8C_3NC_4^2+5)^{(t-1)K+k-2} \\ &\quad+8\varepsilon(1+\zeta)^{t-1}C_4C_3N(8C_3NC_4^2+5)^{2(t-1)K+2k-4}+1\Big).\end{aligned}$$

The claim in the lemma now follows by replacing the bound on ε in the parentheses of the above inequality, followed by some manipulations. ∎

The next lemma shows that if we know bounds on errors in $\{\widehat{E}_{i,k,R}^{(t)}\}_{k=1}^K$ for any t then we can bound the error in $\widehat{E}_{i,1,R}^{(t+1)}$.

Lemma 4. *Let $\Omega_{i,k}^{(t)}$, $\widetilde{\Omega}_{i,k}^{(t)}$, ε and ζ be as defined in Theorem 3, also perform T_c consensus iterations as given in Theorem 3. Now fix δ_d as in Theorem 1 and suppose (i) P1–P3 are satisfied, (ii) $\Omega_{i,k}^{(t+1)} = \widetilde{\Omega}_{i,k}^{(t+1)}$, (iii) $\Omega_{i,k}^{(t)} = \widetilde{\Omega}_{i,k}^{(t)}$, and (iv) $\varepsilon \le \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)}}$, then for any $t \in \{1,\cdots,T_d-1\}$, and for all $k \in \{1,\cdots,K\}$ and $i \in \{1,\cdots,N\}$, if $\|B_{i,k,R}^{(t)}\|_2 \le \varepsilon(1+\zeta)^{t-1}C_4(8C_3C_4^2N+5)^{(t-1)K+k-2}$ then, $\|B_{i,1,R}^{(t+1)}\|_2 \le \varepsilon(1+\zeta)^tC_4(8C_3C_4^2N+5)^{tK-1}$.*

Proof: The error in $\widehat{E}_{i,1,R}^{(t+1)}$ is due to error in dictionary in the previous iteration t and sparse coding at the start of iteration $(t+1)$. Specifically, $B_{i,1}^{(t+1)} = E_{i,1}^{(t+1)} - \widehat{E}_{i,1}^{(t+1)} = Y_i - \sum_{j=2}^K d_j^{(t)}x_{i,j,T}^{(t+1)} - Y_i + \sum_{j=2}^K \widehat{d}_{i,j}^{(t)}\widetilde{x}_{i,j,T}^{(t+1)}$. It then follow that $\|B_{i,1}^{(t+1)}\|_2 \le \sum_{j=2}^K \|\widehat{d}_{i,j}^{(t)}\widetilde{x}_{i,j,T}^{(t+1)} - d_j^{(t)}x_{i,j,T}^{(t+1)}\|_2 \le \sum_{j=1}^K \|\widehat{d}_{i,j}^{(t)}\widetilde{x}_{i,j,T}^{(t+1)} - d_j^{(t)}x_{i,j,T}^{(t+1)}\|_2$. In reality we are interested in finding a bound on $\|B_{i,1,R}^{(t+1)}\|_2$. But since $\Omega_{i,k}^{(t+1)} = \widetilde{\Omega}_{i,k}^{(t+1)}$ we can define $B_{i,1,R}^{(t+1)}$ as $B_{i,1,R}^{(t+1)} = \left(\sum_{j=2}^K\left(Y_i - \widehat{d}_{i,j}^{(t)}\widetilde{x}_{i,j,T}^{(t+1)}\right) - \sum_{j=2}^K\left(Y_i - d_j^{(t)}x_{i,j,T}^{(t+1)}\right)\right)\Omega_{i,1}^{(t+1)}$. It can be seen from this definition that $B_{i,1,R}^{(t+1)}$ is a submatrix of $B_{i,1}^{(t+1)}$, which implies

$$\|B_{i,1,R}^{(t+1)}\|_2 \le \|B_{i,1}^{(t+1)}\|_2 \le \sum_{j=1}^K \|\widehat{d}_{i,j}^{(t)}\widetilde{x}_{i,j,T}^{(t+1)} - d_j^{(t)}x_{i,j,T}^{(t+1)}\|_2. \quad (20)$$

Now, defining $\widehat{d}_{i,j}^{(t)} = d_j^{(t)} + e_{i,j}^{(t)}$, where $e_{i,j}^{(t)}$ denotes the error in dictionary atom $d_j^{(t)}$, and substituting this in (20) we get

$$\begin{aligned}&\|B_{i,1,R}^{(t+1)}\|_2 \\ &\le K\max_j\left(\|d_j^{(t)}\widetilde{x}_{i,j,T}^{(t+1)} - d_j^{(t)}x_{i,j,T}^{(t+1)}\|_2 + \|e_{i,j}\widetilde{x}_{i,j,T}^{(t+1)}\|_2\right) \\ &\le K\max_j\left(\|\widetilde{x}_{i,j,T}^{(t+1)} - x_{i,j,T}^{(t+1)}\|_2 + \|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2\|\widetilde{x}_{i,j,T}^{(t+1)}\|_2\right) \\ &= K\max_j\Big(\|\widetilde{x}_{i,j,T}^{(t+1)} - x_{i,j,T}^{(t+1)}\|_2 \\ &\quad+\|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2\|\widetilde{x}_{i,j,T}^{(t+1)} + x_{i,j,T}^{(t+1)} - x_{i,j,T}^{(t+1)}\|_2\Big) \\ &\le K\max_j\Big(\|\widetilde{x}_{i,j,T}^{(t+1)} - x_{i,j,T}^{(t+1)}\|_2(1+\|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2) \\ &\quad+\|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2\|x_{i,j,T}^{(t+1)}\|_2\Big). \quad (21)\end{aligned}$$

Now, let $X^{(t+1)} = \begin{bmatrix}X_1^{(t+1)} & X_2^{(t+1)} & \dots & X_N^{(t+1)}\end{bmatrix} \in \mathbb{R}^{K\times S}$ be the sparse coding matrix associated with the centralized K-SVD (see, e.g, Sec III-A). Notice that $x_{i,j,T}^{(t+1)}$ is the j^{th} row of $X_i^{(t+1)}$. It then follows that

$$\|x_{i,j,T}^{(t+1)}\|_2 \le \sqrt{S_i}\|X_i^{(t+1)}\|_{\max} \le \sqrt{S_i}\|X_i^{(t+1)}\|_1.$$

We therefore obtain under P1 that $\|x_{i,j,T}^{(t+1)}\|_2 \le \sqrt{S_{\max}}\eta_{\tau,\max}$. Next, using the bound on $\|B_{i,k,R}^{(t)}\|_2$ and applying Lemma 3, we get $\|\widehat{d}_{i,k}^{(t)}\widehat{d}_{i,k}^{(t)^\mathsf{T}} - d_k^{(t)}d_k^{(t)^\mathsf{T}}\|_2 \le \varepsilon(1+\zeta)^{t-1}C_4(8C_3NC_4^2+5)^{(t-1)K+k-1}$. Now, under the assumption that both cloud K-SVD and centralized K-SVD use the same d_{ref}, we have $\widehat{d}_{i,k}^{(t)^\mathsf{T}}d_k^{(t)} \ge 0$ and therefore it follows from Lemma 7 in Appendix D that

$$\begin{aligned}\|\widehat{d}_{i,k}^{(t)} - d_k^{(t)}\|_2 &\le \varepsilon\sqrt{2}(1+\zeta)^{t-1}C_4(8C_3NC_4^2+5)^{(t-1)K+k-1} \\ &\overset{(a)}{\le} \sqrt{2}\delta_d \overset{(b)}{\le} 1, \quad (22)\end{aligned}$$

where (a) follows from the assumption on ε and (b) is true for any fixed δ_d as defined in Theorem 1. Using this bound we can write

$$\begin{aligned}\|D^{(t)} - \widehat{D}_i^{(t)}\|_2 &\le \|D^{(t)} - \widehat{D}_i^{(t)}\|_F = \sqrt{\sum_{j=1}^K\|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2^2} \\ &\le \sqrt{K}\max_{j\in\{1,\cdots,K\}}\|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2 \\ &\le \sqrt{2K}(1+\zeta)^{t-1}\varepsilon C_4(8C_3NC_4^2+5)^{tK-1}. \quad (23)\end{aligned}$$

Furthermore, using lemma assumption on ε we get

$$\|D^{(t)} - \widehat{D}_i^{(t)}\|_2 \le \sqrt{2K}\delta_d = \min\left\{\sqrt{K}, \frac{C_1^2\tau_{\min}}{44}\right\}. \quad (24)$$

We can now use (24) and Proposition 2 in Appendix D to bound $\|x_{i,j,T}^{(t+1)} - \widetilde{x}_{i,j,T}^{(t+1)}\|_2$ in (21). Notice that Proposition 2 assumes the error in dictionary to be smaller than $\frac{C_1^2\tau_{\min}}{44}$, which is satisfied by (24). Other assumptions of Proposition 2 are satisfied due to P1 and P2. Therefore, we get $\forall\ i \in \{1,\dots,N\}$ and $j \in \{1,\dots,S_i\}$,

$$\|x_{i,j}^{(t+1)} - \widetilde{x}_{i,j}^{(t+1)}\|_2 \le \frac{3\sqrt{T_0}}{\tau_{\min}C_2}\|D^{(t)} - \widehat{D}_i^{(t)}\|_2. \quad (25)$$

Now defining $X_i^{(t+1)}$ and $\widetilde{X}_i^{(t+1)}$ as before, we note that

$$\begin{aligned}&\|x_{i,j,T}^{(t+1)} - \widetilde{x}_{i,j,T}^{(t+1)}\|_2 \\ &\le \sqrt{S_{\max}}\|X_i^{(t+1)} - \widetilde{X}_i^{(t+1)}\|_{\max} \\ &\le \sqrt{S_{\max}}\max_{j\in\{1,\dots,S_i\}}\|x_{i,j}^{(t+1)} - \widetilde{x}_{i,j}^{(t+1)}\|_2 \\ &\le \frac{3\sqrt{2KS_{\max}T_0}}{\tau_{\min}C_2}\varepsilon(1+\zeta)^{t-1}C_4(8C_3NC_4^2+5)^{tK-1}, \quad (26)\end{aligned}$$

where the last inequality follows from (25) and (23). Now using bounds on $\|x_{i,j,T}^{(t+1)}\|_2$ and (26) we get the following from (21):

$$\begin{aligned}
&\|B_{i,1,R}^{(t+1)}\|_2 \\
&\overset{(c)}{\leq} 2K \max_j \|\widetilde{x}_{i,j,T}^{(t+1)} - x_{i,j,T}^{(t+1)}\|_2 + \max_j \|\widehat{d}_{i,j}^{(t)} - d_j^{(t)}\|_2 \|x_{i,j,T}^{(t)}\|_2 \\
&\overset{(d)}{\leq} 2K \frac{3\sqrt{S_{\max} T_0}}{\tau_{\min} C_2} \sqrt{2K} \varepsilon (1+\zeta)^{t-1} C_4 (8C_3 N C_4^2 + 5)^{tK-1} \\
&\quad + \varepsilon \sqrt{2} (1+\zeta)^{t-1} C_4 (8C_3 N C_4^2 + 5)^{tK-1} \sqrt{S_{\max}} \eta_{\tau,\max} \\
&\overset{(e)}{\leq} \varepsilon (1+\zeta)^t C_4 (8C_3 N C_4^2 + 5)^{tK-1}.
\end{aligned}$$

Here, (c)–(d) follow by application of (22) and (23), and (e) is by definition of ζ. ∎

The last lemma that we need bounds $\|B_{i,k+1,R}^{(t)}\|_2$ when we have a bound on $\|B_{i,k,R}^{(t)}\|_2$.

Lemma 5. *Let $\Omega_{i,k}^{(t)}$, $\widetilde{\Omega}_{i,k}^{(t)}$, ε and ζ be as defined in Theorem 3, also perform T_c consensus iterations as given in Theorem 3. Now fix δ_d as in Theorem 1, and suppose (i) P1–P3 are satisfied, (ii) $\Omega_{i,k}^{(t)} = \widetilde{\Omega}_{i,k}^{(t)}$, and (iii) $\varepsilon \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4^2(8C_3NC_4^2+5)^{2(T_dK-2)}}$. For any fixed $k \in \{1,\cdots,K\}$, $t \in \{1,\cdots,T_d\}$, and all $i \in \{1,\cdots,N\}$, if $\|B_{i,k,R}^{(t)}\|_2 \leq \varepsilon(1+\zeta)^{t-1} C_4 (8C_3 C_4^2 N + 5)^{(t-1)K+k-2}$ then $\|B_{i,k+1,R}^{(t)}\|_2 \leq \varepsilon(1+\zeta)^{t-1} C_4 (8C_3 C_4^2 N + 5)^{(t-1)K+k-1}$.*

Proof: Recall once again that we can write

$$\begin{aligned}
B_{i,k+1,R}^{(t)} &= \widehat{E}_{i,k+1,R}^{(t)} - E_{i,k+1,R}^{(t)} \\
&= \sum_{j=k+2}^{K} \left(d_j^{(t-1)} x_{i,j,R}^{(t)} - \widehat{d}_{i,j}^{(t-1)} \widetilde{x}_{i,j,R}^{(t)} \right) \\
&\quad - \sum_{j=1}^{k} \left(\widehat{d}_{i,j}^{(t)} \widehat{x}_{i,j,R}^{(t)} - d_j^{(t)} x_{i,j,R}^{(t)} \right),
\end{aligned}$$

now using relation $\widehat{x}_{i,k,R}^{(t)} = \widehat{d}_{i,k}^{(t)\mathsf{T}} \widehat{E}_{i,k,R}^{(t)}$ and doing some rearrangements we get,

$$\begin{aligned}
B_{i,k+1,R}^{(t)} &= \widehat{d}_{i,k}^{(t)} \widehat{d}_{i,k}^{(t)\mathsf{T}} \widehat{E}_{i,k,R}^{(t)} - d_k^{(t)} d_k^{(t)\mathsf{T}} E_{i,k,R}^{(t)} \\
&\quad - \left(d_{k+1}^{(t-1)} x_{i,k+1,R}^{(t)} - \widehat{d}_{i,k+1}^{(t-1)} \widetilde{x}_{i,k+1,R}^{(t)} \right) + B_{i,k,R}^{(t)}.
\end{aligned}$$

It then follows that

$$\begin{aligned}
&\|B_{i,k+1,R}^{(t)}\|_2 \\
&\leq \|B_{i,k,R}^{(t)}\|_2 + \left\| \widehat{d}_{i,k}^{(t)} \widehat{d}_{i,k}^{(t)\mathsf{T}} (E_{i,k,R}^{(t)} + B_{i,k,R}^{(t)}) - d_k^{(t)} d_k^{(t)\mathsf{T}} E_{i,k,R}^{(t)} \right\|_2 \\
&\quad + \left\| d_{k+1}^{(t-1)} x_{i,k+1,R}^{(t)} - \widehat{d}_{i,k+1}^{(t-1)} \widetilde{x}_{i,k+1,R}^{(t)} \right\|_2 \\
&\overset{(a)}{\leq} 2\|B_{i,k,R}^{(t)}\|_2 + \|\widehat{d}_{i,k}^{(t)} \widehat{d}_{i,k}^{(t)\mathsf{T}} - d_k^{(t)} d_k^{(t)\mathsf{T}}\|_2 C_4 \\
&\quad + \left\| d_{k+1}^{(t-1)} x_{i,k+1,R}^{(t)} - \widehat{d}_{i,k+1}^{(t-1)} \widetilde{x}_{i,k+1,R}^{(t)} \right\|_2 \\
&\overset{(b)}{\leq} \varepsilon C_4 (1+\zeta)^{t-1} \Big((8C_3 N C_4^2 + 5)^{(t-1)K+k-2} (8C_3 N C_4^2 + 3) \\
&\quad + \varepsilon 8 C_3 N C_4^2 (1+\zeta)^{t-1} (8C_3 N C_4^2 + 5)^{2(t-1)K+2k-4} \\
&\quad + \frac{1}{(1+\zeta)^{t-1}} \Big).
\end{aligned}$$

Here (a) is due to the fact that $E_{i,k,R}^{(t)}$ is a submatrix of $E_{i,k}^{(t)}$ and the definition of C_4 in (7), (b) is obtained by applying (19), using assumption on $\|B_{i,k,R}^{(t)}\|_2$ and finally using the same procedure as in Lemma 4 after (20) to bound $\sum_{j=1}^{K} \left\| d_j^{(t-1)} x_{i,j,T,R}^{(t)} - \widehat{d}_{i,j}^{(t-1)} \widetilde{x}_{i,j,T,R}^{(t)} \right\|_2$. The proof of the lemma now follows by using the assumption on ε and some algebraic manipulations. ∎

The proof of Theorem 3 now can be given by combining Lemmas 2– 5. Since these lemmas require the supports of both centralized and distributed problems to be the same, the main challenge in proving Theorem 3 lies in showing this fact.

Proof of Theorem 3: We will prove this theorem by mathematical induction over t. To be specific, we will prove the following two cases:

1) For base case, we will show that the claim holds for $\|B_{i,k,R}^{(1)}\|_2 \; \forall k \in \{1,2,\cdots,K\}$.
2) For induction step we assume that for any $q \in \{1,2,\cdots,T_d-1\}$ the claim is true for $\|B_{i,k,R}^{(q)}\|_2 \; \forall k \in \{1,2,\cdots,K\}$ and $\Omega_{i,k}^{(q)} = \widetilde{\Omega}_{i,k}^{(q)}$. Then we need to show that $\Omega_{i,k}^{(q+1)} = \widetilde{\Omega}_{i,k}^{(q+1)}$ and claim holds for $\|B_{i,k,R}^{(q+1)}\|_2 \; \forall k \in \{1,2,\cdots,K\}$.

Base case: $t = 1 \; \forall \; k \in \{1,2,\cdots,K\}$ To prove the base case, we will do mathematical induction over k by fixing $t = 1$. Hence, the first thing we need to prove is that the bound is true for $\|B_{i,1,R}^{(1)}\|_2$. Since both cloud K-SVD and Centralized K-SVD start with the same initial dictionary, we have $d_j^{(0)} = \widehat{d}_{i,j}^{(0)}, \; \forall \; j \in \{1,2,\cdots,K\}$. Therefore, we get $\Omega_{i,j}^{(1)} = \widetilde{\Omega}_{i,j}^{(1)}, \; \forall \; j \in \{1,2,\cdots,K\}$. It then follows that $B_{i,1,R}^{(1)} = E_{i,1,R}^{(1)} - \widehat{E}_{i,1,R}^{(1)} = \sum_{j=1}^{K-1} \left(d_j^{(0)} x_{i,j,T}^{(1)} \Omega_{i,j}^{(1)} - \widehat{d}_j^{(0)} \widetilde{x}_{i,j,T}^{(1)} \widetilde{\Omega}_{i,j}^{(1)} \right) = 0$, thereby proving the claim.

Next, for induction argument we fix $k = p \in \{1,\ldots,K-1\}$ for $t = 1$. Then we need to show that it holds for $k = p+1$. Using the induction assumption we have $\|B_{i,p,R}^{(1)}\|_2 \leq \varepsilon C_4 (8C_3 N C_4^2 + 5)^{p-2}$. Since $\Omega_{i,j}^{(1)} = \widetilde{\Omega}_{i,j}^{(1)}$, we have $B_{i,p+1,R}^{(1)} = \widehat{E}_{i,p+1,R}^{(1)} - E_{i,p+1,R}^{(1)}$. This results in

$$\begin{aligned}
&\|B_{i,p+1,R}^{(1)}\|_2 \\
&= \| \sum_{j=p+2}^{K} \left(d_j^{(0)} x_{i,j,R}^{(1)} - \widehat{d}_{i,j}^{(0)} \widetilde{x}_{i,j,R}^{(1)} \right) \\
&\quad - \sum_{j=1}^{p} \left(\widehat{d}_{i,j}^{(1)} \widehat{x}_{i,j,R}^{(1)} - d_j^{(1)} x_{i,j,R}^{(1)} \right) \|_2 \\
&\overset{(a)}{=} \| - \sum_{j=1}^{p} \left(\widehat{d}_{i,j}^{(1)} \widehat{x}_{i,j,R}^{(1)} - d_j^{(1)} x_{i,j,R}^{(1)} \right) \|_2 \\
&= \|\widehat{d}_{i,p}^{(1)} \widehat{x}_{i,p,R}^{(1)} - d_p^{(1)} x_{i,p,R}^{(1)} + \sum_{j=1}^{p-1} \left(\widehat{d}_{i,j}^{(1)} \widehat{x}_{i,j,R}^{(1)} - d_j^{(1)} x_{i,j,R}^{(1)} \right) \|_2 \\
&= \|\widehat{d}_{i,p}^{(1)} \widehat{x}_{i,p,R}^{(1)} - d_p^{(1)} x_{i,p,R}^{(1)} + B_{i,p,R}^{(1)}\|_2, \qquad (27)
\end{aligned}$$

where (a) is true because $d_j^{(0)} x_{i,j,R}^{(1)} - \widehat{d}_{i,j}^{(0)} \widetilde{x}_{i,j,R}^{(0)} = 0$. Substituting $\widehat{x}_{i,p,R}^{(1)} = \widehat{d}_{i,p}^{(1)\mathsf{T}} \widehat{E}_{i,p,R}^{(1)}$, we get

$$\begin{aligned}
&\|B_{i,p+1,R}^{(1)}\|_2 \\
&\leq 2\|B_{i,p,R}^{(1)}\|_2 + \|\widehat{d}_{i,p}^{(1)} \widehat{d}_{i,p}^{(1)\mathsf{T}} - d_p^{(1)} d_p^{(1)\mathsf{T}}\|_2 \|E_{i,p,R}^{(1)}\|_2
\end{aligned}$$

$$\overset{(b)}{\leq} 2\|B^{(1)}_{i,p,R}\|_2 + \|\widehat{d}^{(1)}_{i,p}\widehat{d}^{(1)^\intercal}_{i,p} - d^{(1)}_p d^{(1)^\intercal}_p\|_2 C_4$$
$$\overset{(c)}{\leq} 2\|B^{(1)}_{i,p,R}\|_2 + C_4\left(8C_3N\max_i\left(\|B^{(1)}_{i,p,R}\|_2 C_4 + \|B^{(1)}_{i,p,R}\|_2^2\right) + \varepsilon\right)$$
$$\overset{(d)}{\leq} \varepsilon C_4\left((8C_3NC_4^2+5)^{p-2}(8C_3NC_4^2+2) + \varepsilon 8C_3NC_4^2(8C_3NC_4^2+5)^{2p-4}+1\right).$$

Here, (b) is true since $E^{(t)}_{i,k,R}$ is a submatrix of $E^{(t)}_{i,k}$ and due to the definition of C_4 in (7), (c) is due to (19) and (d) follows from using the bound on $\|B^{(1)}_{i,p,R}\|_2$ and some manipulations. Now using the assumption on ε, we get $\|B^{(1)}_{i,p+1,R}\|_2 \leq \varepsilon C_4(8C_3NC_4^2+5)^{p-1}$.

Induction step: Bound on $\|B^{(q)}_{i,k,R}\|_2$ holds for $t = q \in \{1,\ldots,T_d-1\}$ and $\forall\, k \in \{1,2,\cdots,K\}$.

We need to show the bound holds for $\|B^{(q+1)}_{i,k,R}\|_2\ \forall\ k \in \{1,\ldots,K\}$. To show this, we will be using induction argument over k by fixing $t = q+1$. As base case we bound $\|B^{(q+1)}_{i,1,R}\|_2$. To bound $\|B^{(q+1)}_{i,1,R}\|_2$, we will be using Lemma 4, which assumes $\widetilde{\Omega}^{(q+1)}_{i,1} = \Omega^{(q+1)}_{i,1}$. Using the induction assumptions, we get the following bound on error in dictionary $\widehat{D}^{(q)}_i$ using Lemma 3 and performing same steps as we carried out in Lemma 4 to get (22) and (23): $\|D^{(q)} - \widehat{D}^{(q)}_i\|_2 \leq \varepsilon\sqrt{2K}C_4(8C_3NC_4^2+5)^{(q-1)K+k-2}$. Using the assumption on ε, we then have $\|D^{(q)} - \widehat{D}^{(q)}_i\|_2 \leq \delta_d\sqrt{2K}$. It then follows from arguments similar to the ones made in Lemma 4 that $\Omega^{(q+1)}_{i,1} = \widetilde{\Omega}^{(q+1)}_{i,1}$. We can now use Lemma 4 to bound $\|B^{(q+1)}_{i,1,R}\|_2 \leq \varepsilon(1+\zeta)^q C_4(8NC_3C_4^2+5)^{qK-1}$. Having proved the base case, we now suppose that the claim is true for some $k = p \in \{1,\ldots,K-1\}$. We then need to show it holds for $\|B^{(q+1)}_{i,p+1,R}\|_2$. That claim, however, simply follows from Lemma 5. This concludes the proof of theorem. ∎

Appendix C
Proof of Theorem 1

To prove Theorem 1 we need an upper bound on error in matrices $\widehat{E}^{(t)}_{i,k,R}$, which is given by Theorem 3. Applying Theorem 3 to get a bound on the error in dictionary atom $\widehat{d}^{(t)}_{i,k}$ is a trivial task. But before using Theorem 3, we need to show that our assumption on ε is indeed satisfied. In the following, we will prove that the assumption on ε is satisfied if we perform T_p power method and T_c consensus iterations that are given according to the statement of Theorem 1.

Proof of Theorem 1: After T_d iterations of cloud K-SVD, error in any k^{th} dictionary atom $\widehat{d}^{(T_d)}_{i,k}$ at site i is a function of the error in $\widehat{E}^{(T_d)}_{i,k,R}$. Specifically, notice from (19) that we can write

$$\|d^{(T_d)}_k d^{(T_d)^\intercal}_k - \widehat{d}^{(T_d)}_{i,k}\widehat{d}^{(T_d)^\intercal}_{i,k}\|_2 \leq 8NC_3\max_i\left(\|B^{(T_d)}_{i,k,R}\|_2 C_4 + \|B^{(T_d)}_{i,k,R}\|_2^2\right) + \varepsilon. \quad (28)$$

We can now upper bound $\|B^{(T_d)}_{i,k,R}\|_2$ in (28) using Theorem 3, but we first need to show that the statement of Theorem 1 implies the assumption on ε in Theorem 3 is satisfied. That is, we need to show $\varepsilon \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4(8C_3NC_4^2+5)^{2(T_dK-2)}}$. Recall that by definition $\varepsilon = \mu\nu^{T_p} + 4\epsilon^{3T_p}$. Substituting this, we must show that

$$\mu\nu^{T_p} + 4\epsilon^{3T_p} \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4(8C_3NC_4^2+5)^{2(T_dK-2)}}.$$

Since $\nu > 0$ and $\epsilon > 0$, therefore, $\mu\nu^{T_p} + 4\epsilon^{3T_p} < \mu(\nu + 4\epsilon^3)^{T_p}$. It is therefore sufficient to show that $\mu(\nu+4\epsilon^3)^{T_p} \leq \frac{\delta_d}{8N\sqrt{n}C_3(1+\zeta)^{T_d-1}C_4(8C_3NC_4^2+5)^{2(T_dK-2)}}$ for our selected values of T_p and T_c. Showing that, however, is a simple exercise and is left out for brevity. It therefore follows from Theorem 3 that $\|d^{(T_d)}_k d^{(T_d)^\intercal}_k - \widehat{d}^{(T_d)}_k\widehat{d}^{(T_d)^\intercal}_k\|_2 \leq \varepsilon(1+\zeta)^{T_d-1}(8C_3NC_4^2+5)^{(T_d-1)K+k-1}$. Substituting the upper bound on ε, we get $\|d^{(T_d)}_k d^{(T_d)^\intercal}_k - \widehat{d}^{(T_d)}_k\widehat{d}^{(T_d)^\intercal}_k\|_2 \leq \delta_d$. ∎

Appendix D
Other results

In this appendix, we collect some supporting results that are used in the proofs of our main results.

Lemma 6 (Perturbation of singular values)**.** *Let D_2 be a perturbed version of dictionary D_1 such that $\|D_1 - D_2\|_2 \leq \epsilon_2$ and let Σ_{T_0} be as defined in Section IV-A. Then assuming $\min_{\mathcal{I}\in\Sigma_{T_0}}\sigma_{T_0}\left(D_{1_{|\mathcal{I}}}\right) \geq \sqrt{C_2'} > \epsilon_2$, we have $\min_{\mathcal{I}\in\Sigma_{T_0}}\sigma_{T_0}\left(D_{2_{|\mathcal{I}}}\right) \geq \sqrt{C_2'} - \epsilon_2$.*

Proof: Using [56, Theorem 1], perturbation in T_0^{th} singular value of $D_{1_{|\mathcal{I}}}$ can be bounded as $|\sigma_{T_0}\left(D_{1_{|\mathcal{I}}}\right) - \sigma_{T_0}\left(D_{2_{|\mathcal{I}}}\right)| \leq \|D_{1_{|\mathcal{I}}} - D_{2_{|\mathcal{I}}}\|_2 \leq \|D_1 - D_2\|_2 \leq \epsilon_2$. Using reverse triangular inequality, we therefor get $\forall\mathcal{I}\in\Sigma_{T_0}, \epsilon_2 \geq |\sigma_{T_0}\left(D_{1_{|\mathcal{I}}}\right)| - |\sigma_{T_0}\left(D_{2_{|\mathcal{I}}}\right)| \geq \sqrt{C_2'} - |\sigma_{T_0}\left(D_{2_{|\mathcal{I}}}\right)|$, which leads to the claimed result. ∎

Proposition 2 (Stability of sparse coding)**.** *[57, Theorem 1] Let D_2 be a perturbed version of dictionary D_1 such that $\|D_1 - D_2\|_2 \leq \epsilon_2$. Given any sample $y \in \mathbb{R}^n$, suppose sparse codes $x \in \mathbb{R}^K$ and $\widehat{x} \in \mathbb{R}^K$ are computed by solving the lasso problem* (5) *using D_1 and D_2, respectively. Next, let $\min_{j\notin supp(x)}\tau - |\langle d_{1,j}, y - D_1x|| > C_1$, where $d_{1,j}$ denotes the j^{th} atom of D_1, and suppose D_1 satisfies P2. Then, as long as $\epsilon_2 \leq \frac{C_1^2\tau}{44}$, we have that $supp(x) = supp(\widehat{x})$ and $\|x - \widehat{x}\|_2 \leq \frac{3\|D_1-D_2\|_2\sqrt{T_0}}{\tau C_2}$, where $T_0 = |supp(x)|$.*

Note that [57, Theorem 1] also requires D_2 to satisfy P2. Proposition 2 in its current form, however, is a simple consequence of [57, Theorem 1] and Lemma 6.

Proposition 3 (Perturbation of principal eigenvector)**.** *[26, Chap. 8] Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and define $\widehat{A} = A + E$ to be a perturbed, but symmetric version of A. Define $Q = \begin{bmatrix} q_1 & | & Q_2\end{bmatrix}$ to be an orthogonal matrix comprising eigenvectors of A, where q_1 denotes the principal eigenvector of A. Next, define $Q^\intercal AQ = \begin{bmatrix}\lambda & 0\\ 0 & \Lambda_2\end{bmatrix}$ and $Q^\intercal EQ = \begin{bmatrix}\epsilon & e^\intercal\\ e & E_{22}\end{bmatrix}$. Then, using $eig(\Lambda_2)$ to denote the $(n-1)$ smallest eigenvalues of A, it follows that if $g = \min_{\varrho\in eig(\Lambda_2)}|\lambda - \varrho| > 0$, and $\|E\|_F \leq \frac{g}{5}$ then there exists $p \in \mathbb{R}^{n-1}$ satisfying $\|p\|_2 \leq \frac{4}{g}\|e\|_2$, such that $\widehat{q}_1 = \frac{q_1+Q_2p}{\sqrt{1+p^\intercal p}}$ is a unit 2-norm principal eigenvector for $\widehat{A}$. Moreover, $\|q_1q_1^\intercal - \widehat{q}_1\widehat{q}_1^\intercal\|_2 \leq \frac{4}{g}\|e\|_2$.*

Lemma 7 (Errors in vectors and their outerproducts)**.** *For two unit ℓ_2-norm vectors u and v if $\|uu^\mathsf{T} - vv^\mathsf{T}\|_2 \leq \epsilon$ and $u^\mathsf{T}v \geq 0$ then $\|u - v\|_2 \leq \sqrt{2}\epsilon$.*

Proof: Let $\theta = \angle(u, v)$ and notice that $\|uu^\mathsf{T} - vv^\mathsf{T}\|_2 = \sin\theta$. This implies $1 - \cos^2\theta = \sin^2\theta = \|uu^\mathsf{T} - vv^\mathsf{T}\|_2^2 \leq \epsilon^2$. Since u and v are unit norm and $u^\mathsf{T}v \geq 0$, we can write $\cos\theta = u^\mathsf{T}v$. It then follows that $1 - u^\mathsf{T}v \leq \frac{\epsilon^2}{1+u^\mathsf{T}v} < \epsilon^2$. The claim follows by noting that $\|u - v\|_2 = \sqrt{2(1 - u^\mathsf{T}v)}$. ∎

REFERENCES

[1] H. Raja and W. U. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Proc. 51st Allerton Conf.*, 2013.

[2] H. Raja and W. U. Bajwa, "A convergence analysis of distributed dictionary learning based on the K-SVD algorithm," in *Proc. IEEE Intl. Symp. Information Theory (ISIT)*, 2015.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Edu. Psych.*, vol. 6, no. 24, pp. 417–441, Sep. 1933.

[4] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE PAMI*, vol. 27, pp. 1945–1959, Dec. 2005.

[5] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Intl. J. Comput. Vision*, Dec. 2012.

[6] 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.

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

[8] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[9] M. Gastpar, P.-L. Dragotti, and M. Vetterli, "The distributed Karhunen–Loève transform," *IEEE Trans. Inform. Theory*, , Dec. 2006.

[10] M. Nokleby and W. U. Bajwa, "Resource tradeoffs in distributed subspace tracking over the wireless medium," in *Proc. 1st IEEE GlobalSIP, Symp. on Netw. Theory*, Austin, TX, Dec. 2013, pp. 823–826.

[11] L. Li, A. Scaglione, and J. H. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 725–738, 2011.

[12] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. IEEE 42nd Asilomar Conf. on Signals, Syst. and Comput.*, 2008, pp. 1722–1726.

[13] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *Proc. IEEE, SPAWC*, 2010, pp. 1–5.

[14] R. Tron and R. Vidal, "Distributed computer vision algorithms through distributed averaging," in *Proc. IEEE CVPR*, 2011, pp. 57–63.

[15] S. Yoon and V. Pavlovic, "Distributed probabilistic learning for camera networks with missing data," in *Proc. NIPS*, 2012.

[16] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Trans. Signal Process.*, vol. 63, Feb 2015.

[17] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE 5th CAMSAP*, 2013, pp. 133–136.

[18] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2528–2541, June 2014.

[19] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *IEEE Int. Workshop on Mach. Learning for Signal Process., MLSP*. 2014, pp. 1–6.

[20] J. Chen, C. Richard, and A. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, Aug 2014.

[21] Y. Lu and M. Do, "A theory for sampling signals from a union of subspaces," *IEEE Trans. Signal Process.*, vol. 56, no. 6, Jun. 2008.

[22] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intel.*, Aug. 1996.

[23] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[24] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[25] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. ACM 26th ICML*, 2009, pp. 689–696.

[26] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 2012.

[27] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, Jan. 2007.

[28] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Automat. Control*, vol. 24, no. 2, pp. 266–269, Apr. 1979.

[29] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. 29, no. 1, pp. 42–50, Jan. 1984.

[30] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *The JMLR*, 2010.

[31] E. Kokiopoulou and P. Frossard, "Distributed classification of multiple observation sets by consensus," *IEEE Trans. Signal Process.*, 2011.

[32] S. Lee and A. Nedić, "Distributed random projection algorithm for convex optimization," *IEEE J. Sel. Topics Signal Process.*, Apr. 2013.

[33] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.

[34] ——, "DILAND: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, Mar. 2010.

[35] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, 2010.

[36] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[37] M. E. Yildiz, F. Ciaramello, and A. Scaglione, "Distributed distance estimation for manifold learning and dimensionality reduction," in *Proc. IEEE Int. Conf. on Acous., Speech and Signal Process., ICASSP*, 2009.

[38] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *J. Comput. and Syst. Sci.*, vol. 74, no. 1, pp. 70–83, 2008.

[39] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: Theory and design," *Signal Process.*, vol. 80, no. 10, 2000.

[40] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, no. 1, 1998.

[41] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[42] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[43] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control and Optimization*, 2009.

[44] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[45] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Series B (Methodological)*, pp. 267–288, 1996.

[46] J. Ding, L. Chen, and Y. Gu, "Perturbation analysis of orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 2, 2013.

[47] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–451, 2004.

[48] 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. Sel. Topics Signal Process.*, Dec. 2007.

[49] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1341–1344, Jun. 2004.

[50] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Math. Soc., 2009.

[51] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Mixing times for random walks on geometric random graphs." in *Proc. ALENEX/ANALCO*. SIAM, 2005, pp. 240–249.

[52] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[53] T. Wu, A. D. Sarwate, and W. U. Bajwa, "Active dictionary learning for image representation," *in Proc. SPIE Unmanned Systems Technology XVII,* Baltimore, MD, Apr. 21-23, 2015.

[54] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.

[55] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, 2011.

[56] G. W. Stewart, "Perturbation theory for the singular value decomposition," *Technical Report.*, 1998.

[57] N. Mehta and A. G. Gray, "Sparsity-based generalization bounds for predictive sparse coding," in *Proc. 30th ICML*, Atlanta, GA, Jun. 2013.