Learning Stable Multilevel Dictionaries for Sparse Representations

Jayaraman J. Thiagarajan, Karthikeyan Natesan Ramamurthy and Andreas Spanias E-mail: {jjayaram,knatesan,spanias}@asu.edu.

Abstract—Sparse representations using learned dictionaries are being increasingly used with success in several data processing and machine learning applications. The availability of abundant training data necessitates the development of efficient, robust and provably good dictionary learning algorithms. Algorithmic stability and generalization are desirable characteristics for dictionary learning algorithms that aim to build global dictionaries which can efficiently model any test data similar to the training samples. In this paper, we propose an algorithm to learn dictionaries for sparse representations from large scale data, and prove that the proposed learning algorithm is stable and generalizable asymptotically. The algorithm employs a 1-D subspace clustering procedure, the K-hyperline clustering, in order to learn a hierarchical dictionary with multiple levels. We also propose an information-theoretic scheme to estimate the number of atoms needed in each level of learning and develop an ensemble approach to learn robust dictionaries. Using the proposed dictionaries, the sparse code for novel test data can be computed using a low-complexity pursuit procedure. We demonstrate the stability and generalization characteristics of the proposed algorithm using simulations. We also evaluate the utility of the multilevel dictionaries in compressed recovery and subspace learning applications.

I. INTRODUCTION

A. Dictionary Learning for Sparse Representations

SEVERAL types of naturally occurring data have most of their energy concentrated in a small number of features when represented using an linear model. In particular, it has been shown that the statistical structure of naturally occurring signals and images allows for their efficient representation as a sparse linear combination of elementary features [1]. A finite collection of normalized features is referred to as a dictionary. The linear model used for general sparse coding is given by

$$\mathbf{y} = \mathbf{\Psi}\mathbf{a} + \mathbf{n},\tag{1}$$

where $\mathbf{y} \in \mathbb{R}^M$ is the data vector and $\mathbf{\Psi} = [\psi_1 \psi_2 \dots \psi_K] \in \mathbb{R}^{M \times K}$ is the dictionary. Each column of the dictionary, referred to as an atom, is a representative pattern normalized to unit ℓ_2 norm. $\mathbf{a} \in \mathbb{R}^K$ is the sparse coefficient vector and \mathbf{n} is a noise vector whose elements are independent realizations from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

The sparse coding problem is usually solved as

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{a}\|_0 \text{ subj. to } \|\mathbf{y} - \mathbf{\Psi}\mathbf{a}\|_2^2 \le \epsilon,$$
 (2)

where $\|.\|_0$ indicates the ℓ_0 sparsity measure which counts the number of non-zero elements, $\|.\|_2$ denotes the ℓ_2 norm and ϵ

The authors are with the SenSIP Center, School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, 85287.

is the error goal for the representation. The ℓ_1 norm, denoted by $\|.\|_1$, can be used instead of ℓ_0 measure to convexify (2). A variety of methods can be found in the literature to obtain sparse representations efficiently [2]–[5]. The sparse coding model has been successfully used for inverse problems in images [6], and also in machine learning applications such as classification, clustering, and subspace learning to name a few [7]–[16].

The dictionary Ψ used in (2) can be obtained from predefined bases, designed from a union of orthonormal bases [17], or structured as an overcomplete set of individual vectors optimized to the data [18]. A wide range of batch and online dictionary learning algorithms have been proposed in the literature [19]–[27], some of which are tailored for specific tasks. The conditions under which a dictionary can be identified from the training data using an ℓ_1 minimization approach are derived in [28]. The joint optimization problem for dictionary learning and sparse coding can be expressed as [6]

$$\min_{\mathbf{\Psi}, \mathbf{A}} \|\mathbf{Y} - \mathbf{\Psi}\mathbf{A}\|_F^2 \text{ subj. to } \|\mathbf{a}_i\|_0 \le S, \forall i, \|\boldsymbol{\psi}_j\|_2 = 1, \forall j,$$
(3)

where $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_T]$ is a matrix of T training vectors, $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_T]$ is the coefficient matrix, S is the sparsity of the coefficient vector and $\|.\|_F$ denotes the Frobenius norm.

B. Multilevel Learning

In this paper, we propose a hierarchical multilevel dictionary learning algorithm that is implicitly regularized to aid in sparse approximation of data. The proposed multilevel dictionary (MLD) learning algorithm is geared towards obtaining *global* dictionaries for the entire probability space of the data, which are *provably* stable, and generalizable to novel test data. In addition, our algorithm involves simple schemes for learning and representation: a 1-D subspace clustering algorithm (*K-hyperline clustering* [29]) is used to infer atoms in each level, and 1—sparse representations are obtained in each level using a pursuit scheme that employs just *correlate-and-max* operations. In summary, the algorithm creates a sub-dictionary for each level and obtains a residual which is used as the training data for the next level, and this process is continued until a pre-defined stopping criterion is reached.

The primary utility of sparse models with learned dictionaries in data processing and machine learning applications stems from the fact that the dictionary atoms serve as *predictive features*, capable of providing a good representation for some aspect of the test data. From the viewpoint of statistical

learning theory [30], a good predictive model is one that is stable and generalizable, and MLD learning satisfies both these properties. To the best of our knowledge, there is no other dictionary learning method which has been proven to satisfy these properties. Generalization ensures that the learned dictionary can successfully represent test data drawn from the same probability space as the training data, and stability guarantees that it is possible to reliably learn such a dictionary from an arbitrary training set. In other words, the asymptotic stability and generalization of MLD provides theoretical justification for the uniformly good performance of global multilevel dictionaries. We can minimize the risk of overfitting further by choosing a proper model order. We propose a method based on the minimum description length (MDL) principle [31] to choose the optimal model order, which in our case corresponds to the number of dictionary elements in each level. Recently, other approaches have been proposed to choose the best order for a given sparse model using MDL [27], so that the generalization error is minimized. However, the difference in our case is that, in addition to optimizing the model order for a given training set using MDL, we prove that any dictionary learned using MLD is generalizable and stable. Since both generalization and stability are asymptotic properties, we also propose a robust variant of our MLD algorithm using randomized ensemble methods, to obtain an improved performance with test data. Note that our goal is not to obtain dictionaries optimized for a specific task [24], but to propose a general predictive sparse modeling framework that can be suitably adapted for any task.

The dictionary atoms in MLD are structurally regularized, and therefore the hierarchy in representation is imposed implicitly for the novel test data, leading to improved recovery in ill-posed and noise-corrupted problems. Considering dictionary learning with image patches as an example, in MLD the predominant atoms in the first few levels (see Figure 1) always contribute the highest energy to the representation. For natural image data, it is known that the patches are comprised of geometric patterns or stochastic textures or a combination of both [32]. Since the geometric patterns usually are of higher energy when compared to stochastic textures in images, MLD learns the geometric patterns in the first few levels and stochastic textures in the last few levels, thereby adhering to the natural hierarchy in image data. The hierarchical multistage vector quantization (MVQ) [33] is related to MLD learning. The important difference, however, is that dictionaries obtained for sparse representations must assume that the data lies in a union-of-subspaces, and the MVO does not incorporate this assumption. Note that multilevel learning is also different from the work in [34], where multiple sub-dictionaries are designed and one of them is chosen for representing a group of patches.

C. Stability and Generalization in Learning

A learning algorithm is a map from the space of training examples to the hypothesis space of functional solutions. In clustering, the learned function is completely characterized by the cluster centers. Stability of a clustering algorithm implies that the cluster centroids learned by the algorithm are not significantly different when different sets of i.i.d. samples from the same probability space are used for training [35]. When there is a unique minimizer to the clustering objective with respect to the underlying data distribution, stability of a clustering algorithm is guaranteed [36] and this analysis has been extended to characterize the stability of K-means clustering in terms of the number of minimizers [37]. In [38], the stability properties of the K-hyperline clustering algorithm have been analyzed and they have been shown to be similar to those of K-means clustering. Note that all the stability characterizations depend only on the underlying data distribution and the number of clusters, and not on the actual training data itself. Generalization implies that the average empirical training error becomes asymptotically close to the expected error with respect to the probability space of data. In [39], the generalization bound for sparse coding in terms of the number of samples T, also referred to as sample complexity, is derived and in [40] the bound is improved by assuming a class of dictionaries that are nearly orthogonal. Clustering algorithms such as the K-means and the K-hyperline can be obtained by constraining the desired sparsity in (3) to be 1. Since the stability characteristics of clustering algorithms are well understood, employing similar tools to analyze a general dictionary learning framework such as MLD can be beneficial.

D. Contributions

In this paper, we propose the MLD learning algorithm to design global representative dictionaries for image patches. We show that, for a sufficient number of levels, the proposed algorithm converges, and also demonstrate that a multilevel dictionary with a sufficient number of atoms per level exhibits energy hierarchy (Section III-B). Furthermore, in order to estimate the number of atoms in each level of MLD, we provide an information-theoretic approach based on the MDL principle (Section III-C). In order to compute sparse codes for test data using the proposed dictionary, we develop the simple Multilevel Pursuit (MulP) procedure and quantify its computational complexity (Section III-D). We also propose a method to obtain robust dictionaries with limited training data using ensemble methods (Section III-E). Some preliminary algorithmic details and results obtained using MLD have been reported in [41].

Using the fact that the K-hyperline clustering algorithm is stable, we perform stability analysis of the MLD algorithm. For any two sets of i.i.d. training samples from the same probability space, as the number of training samples $T \to \infty$, we show that the dictionaries learned become close to each other asymptotically. When there is a unique minimizer to the objective in each level of learning, this holds true even if the training sets are completely disjoint. However, when there are multiple minimizers for the objective in at least one level, we prove that the learned dictionaries are asymptotically close when the difference between their corresponding training sets is $o(\sqrt{T})$. Instability of the algorithm when the difference between two training sets is $\Omega(\sqrt{T})$, is also shown for the case of multiple minimizers (Section IV-C). Furthermore, we prove the asymptotic generalization of the learning algorithm (Section IV-D).

In addition to demonstrating the stability and the generalization behavior of MLD learning with image data (Sections V-A and V-B), we evaluate its performance in compressed recovery of images (Section V-C). Due to its theoretical guarantees, the proposed MLD effectively recovers novel test images from severe degradation (random projection). Interestingly, the proposed greedy pursuit with robust multilevel dictionaries results in improved recovery performance when compared to ℓ_1 minimization with online dictionaries, particularly at reduced number of measurements and in presence of noise. Furthermore, we perform subspace learning with graphs constructed using sparse codes from MLD and evaluate its performance in classification (Section V-D). We show that the proposed approach outperforms subspace learning with neighborhood graphs as well as graphs based on sparse codes from conventional dictionaries.

II. BACKGROUND

In this section, we describe the K-hyperline clustering, a 1-D subspace clustering procedure proposed in [29], which forms a building block of the proposed dictionary learning algorithm. Furthermore, we briefly discuss the results for stability analysis of K-means and K-hyperline algorithms reported in [35] and [38] respectively. The ideas described in this section will be used in Section IV to study the stability characteristics of the proposed dictionary learning procedure.

A. K-hyperline Clustering Algorithm

The K-hyperline clustering algorithm is an iterative procedure that performs a least squares fit of K 1-D linear subspaces to the training data [29]. Note that the K-hyperline clustering is a special case of general subspace clustering methods proposed in [42], [43], when the subspaces are 1-dimensional and constrained to pass through the origin. In contrast with Kmeans, K-hyperline clustering allows each data sample to have an arbitrary coefficient value corresponding to the centroid of the cluster it belongs to. Furthermore, the cluster centroids are normalized to unit ℓ_2 norm. Given the set of T data samples $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^T$ and the number of clusters K, Khyperline clustering proceeds in two stages after initialization: the cluster assignment and the cluster centroid update. In the cluster assignment stage, training vector \mathbf{y}_i is assigned to a cluster j based on the minimum distortion criteria, $\mathcal{H}(\mathbf{y}_i) = \operatorname{argmin}_i d(\mathbf{y}_i, \boldsymbol{\psi}_i)$, where the distortion measure is

$$d(\mathbf{y}, \boldsymbol{\psi}) = \|\mathbf{y} - \boldsymbol{\psi}(\mathbf{y}^T \boldsymbol{\psi})\|_2^2. \tag{4}$$

In the cluster centroid update stage, we perform singular value decomposition (SVD) of $\mathbf{Y}_j = [\mathbf{y}_i]_{i \in \mathcal{C}_j}$, where $\mathcal{C}_j = \{i | \mathcal{H}(\mathbf{y}_i) = j\}$ contains indices of training vectors assigned to the cluster j. The cluster centroid is updated as the left singular vector corresponding to the largest singular value of the decomposition. This can also be computed using a linear iterative procedure. At iteration t+1, the j^{th} cluster centroid is given by

$$\psi_j^{(t+1)} = \mathbf{Y}_j \mathbf{Y}_j^T \psi_j^{(t)} / \|\mathbf{Y}_j \mathbf{Y}_j^T \psi_j^{(t)}\|_2.$$
 (5)

Usually a few iterations are sufficient to obtain the centroids with good accuracy.

B. Stability Analysis of Clustering Algorithms

Analyzing the stability of unsupervised clustering algorithms can be valuable in terms of understanding their behavior with respect to perturbations in the training set. These algorithms extract the underlying structure in the training data and the quality of clustering is determined by an accompanying cost function. As a result, any clustering algorithm can be posed as an Empirical Risk Minimization (ERM) procedure, by defining a hypothesis class of loss functions to evaluate the possible cluster configurations and to measure their quality [44]. For example, K-hyperline clustering can be posed as an ERM problem over the distortion function class

$$\mathcal{G}_K = \left\{ g_{\mathbf{\Psi}}(\mathbf{y}) = d(\mathbf{y}, \boldsymbol{\psi}_j), j = \underset{l \in \{1, \dots, K\}}{\operatorname{argmax}} |\mathbf{y}^T \boldsymbol{\psi}_l| \right\}. \quad (6)$$

The class \mathcal{G}_K is constructed with functions g_{Ψ} corresponding to all possible combinations of K unit length vectors from the \mathbb{R}^M space for the set Ψ . Let us define the probability space for the data in \mathbb{R}^M as (\mathcal{Y}, Σ, P) , where \mathcal{Y} is the sample space and Σ is a sigma-algebra on \mathcal{Y} , i.e., the collection of subsets of \mathcal{Y} over which the probability measure P is defined. The training samples, $\{\mathbf{y}_i\}_{i=1}^T$, are i.i.d. realizations from this space.

Ideally, we are interested in computing the cluster centroids $\hat{\Psi}$ that minimize the expected distortion $\mathbb{E}[g_{\Psi}]$ with respect to the probability measure P. However, the underlying distribution of the data samples is not known and hence we resort to minimizing the average empirical distortion with respect to the training samples $\{\mathbf{y}_i\}_{i=1}^T$ as

$$g_{\hat{\mathbf{\Psi}}} = \underset{g \in \mathcal{G}_K}{\operatorname{argmin}} \frac{1}{T} \sum_{i=1}^{T} g_{\mathbf{\Psi}}(\mathbf{y}_i). \tag{7}$$

When the empirical averages of the distortion functions in \mathcal{G}_K uniformly converge to the expected values over all probability measures P,

$$\lim_{T \to \infty} \sup_{P} \mathbb{P} \left(\sup_{g_{\Psi} \in \mathcal{G}_K} \left| \mathbb{E}[g_{\Psi}] - \frac{1}{T} \sum_{i=1}^{T} g_{\Psi}(\mathbf{y}_i) \right| > \delta \right) = 0,$$
(8)

for any $\delta>0$, we refer to the class \mathcal{G}_K as uniform Glivenko-Cantelli (uGC). In addition, if the class also satisfies a version of the central limit theorem, it is defined as uniform Donsker [44]. In order to determine if \mathcal{G}_K is uniform Donsker, we have to verify if the covering number of \mathcal{G}_K with respect to the supremum norm, $N_\infty(\gamma,\mathcal{G}_K)$, grows polynomially in the dimensions M [35]. Here, γ denotes the maximum L_∞ distance between an arbitrary distortion function in \mathcal{G}_K , and the function that covers it. For K-hyperline clustering, the covering number is upper bounded by [38, Lemma 2.1]

$$N_{\infty}(\gamma, \mathcal{G}_K) \le \left(\frac{8R^3K + \gamma}{\gamma}\right)^{MK},$$
 (9)

where we assume that the data lies in an M-dimensional ℓ_2 ball of radius R centered at the origin. Therefore, \mathcal{G}_K belongs to the uniform Donsker class.

Stability implies that the algorithm should produce cluster centroids that are not significantly different when different i.i.d. sets from the same probability space are used for training [35]–[37]. Stability is characterized based on the number of minimizers to the clustering objective with respect to the underlying data distribution. A minimizer corresponds to a function $g_{\Psi} \in \mathcal{G}_K$ with the minimum expectation $\mathbb{E}[g_{\Psi}]$. Stability analysis of K-means clustering has been reported in [35], [37]. Though the geometry of K-hyperline clustering is different from that of K-means, the stability characteristics of the two algorithms have been found to be similar [38].

Given two sets of cluster centroids $\Psi = \{\psi_1, \dots, \psi_K\}$ and $\Lambda = \{\lambda_1, \dots, \lambda_K\}$ learned from training sets of T i.i.d. samples each realized from the same probability space, let us define the $L_1(P)$ distance between the clusterings as

$$||g_{\mathbf{\Psi}} - g_{\mathbf{\Lambda}}||_{L_1(P)} = \int |g_{\mathbf{\Psi}}(\mathbf{y}) - g_{\mathbf{\Lambda}}(\mathbf{y})| dP(\mathbf{y}).$$
 (10)

When $T \to \infty$, and \mathcal{G}_K is uniform Donsker, stability in terms of the distortion functions is expressed as

$$||g_{\Psi} - g_{\Lambda}||_{L_1(P)} \xrightarrow{P} 0, \tag{11}$$

where \xrightarrow{P} denotes convergence in probability. This holds true even for Ψ and Λ learned from completely disjoint training sets, when there is a unique minimizer to the clustering objective. When there are multiple minimizers, (11) holds true with respect to a change in $o(\sqrt{T})$ samples between two training sets and fails to hold with respect to a change in $\Omega(\sqrt{T})$ samples [38]. The distance between the cluster centroids themselves is defined as [35]

$$\Delta(\boldsymbol{\Psi}, \boldsymbol{\Lambda}) = \max_{1 \le j \le K} \min_{1 \le l \le K} \left[(d(\boldsymbol{\psi}_j, \boldsymbol{\lambda}_l))^{1/2} + (d(\boldsymbol{\psi}_l, \boldsymbol{\lambda}_j))^{1/2} \right]. \tag{12}$$

Lemma 2.1 ([38]): If the $L_1(P)$ distance between the distortion functions for the clusterings Ψ and Λ is bounded as $\|g_{\Psi} - g_{\Lambda}\|_{L_1(P)} < \mu$, for some $\mu > 0$, and $dP(\mathbf{y})/d\mathbf{y} > C$, for some C > 0, then $\Delta(\Psi, \Lambda) \leq 2\sin(\rho)$ where

$$\rho \le 2\sin^{-1} \left[\frac{1}{r} \left(\frac{\mu}{\hat{C}_{C,M}} \right)^{\frac{1}{M+1}} \right]. \tag{13}$$

Here the training data is assumed to lie outside an M-dimensional ℓ_2 ball of radius r centered at the origin, and the constant $\hat{C}_{C,M}$ depends only on C and M.

When the clustering algorithm is stable according to (11), for admissible values of r, Lemma 2.1 shows that the cluster centroids become arbitrarily close to each other.

III. MULTILEVEL DICTIONARY LEARNING

In this section, we develop the multilevel dictionary learning algorithm, whose algorithmic stability and generalizability will be proved in Section IV. Furthermore, we propose strategies to estimate the number of atoms in each level and make the learning process robust for improved generalization. We also present a simple pursuit scheme to compute representations for novel test data using the MLD.

TABLE I ALGORITHM FOR BUILDING A MULTILEVEL DICTIONARY.

$\mathbf{Y} = [\mathbf{y}_i]_{i=1}^T$, $M \times T$ matrix of training vectors. L, maximum number of levels of the dictionary. K_l , number of dictionary elements in level $l, l = \{1, 2, ..., L\}$. ϵ , error goal of the representation. Output Ψ_l , adapted sub-dictionary for level l. Initialize: l = 1 and $\mathbf{R}_0 = \mathbf{Y}$. $\Lambda_0 = \{i \mid \|\mathbf{r}_{0,i}\|_2^2 > \epsilon, 1 \le i \le T\}, \text{ index of training vectors with }$ squared norm greater than error goal. $\hat{\mathbf{R}}_0 = [\mathbf{r}_{0,i}]_{i \in \Lambda_0} \,.$ while $\Lambda_{l-1} \neq \emptyset$ and $l \leq L$ Initialize: \mathbf{A}_l , coefficient matrix, size $K_l \times M$, all zeros. \mathbf{R}_l , residual matrix for level l, size $M \times T$, all zeros. $\{\mathbf{\Psi}_l, \hat{\mathbf{A}}_l\} = \text{KLC}(\hat{\mathbf{R}}_{l-1}, K_l).$ $\mathbf{R}_l^t = \hat{\mathbf{R}}_{l-1} - \mathbf{\Psi}_l \hat{\mathbf{A}}_l.$ $$\begin{split} \mathbf{r}_{l,i}^i &= \mathbf{r}_{l,j}^i \text{ where } i = \Lambda_{l-1}(j), \ \forall j = 1,..., |\Lambda_{l-1}|.\\ \mathbf{a}_{l,i} &= \hat{\mathbf{a}}_{l,j} \text{ where } i = \Lambda_{l-1}(j), \ \forall j = 1,..., |\Lambda_{l-1}|.\\ \Lambda_l &= \{i \mid \|\mathbf{r}_{l,i}\|_2^2 > \epsilon, 1 \leq i \leq T\}. \end{split}$$ $\hat{\mathbf{R}}_l = \left[\mathbf{r}_{l,i}\right]_{i \in \Lambda_l}.$

A. Algorithm

end

 $l \leftarrow l + 1$.

Input

We denote the MLD as $\Psi = [\Psi_1 \Psi_2 ... \Psi_L]$, and the coefficient matrix as $\mathbf{A} = [\mathbf{A}_1^T \mathbf{A}_2^T ... \mathbf{A}_L^T]^T$. Here, Ψ_l is the sub-dictionary and \mathbf{A}_l is the coefficient matrix for level l. The approximation in level l is expressed as

$$\mathbf{R}_{l-1} = \mathbf{\Psi}_l \mathbf{A}_l + \mathbf{R}_l, \text{ for } l = 1, ..., L, \tag{14}$$

where \mathbf{R}_{l-1} , \mathbf{R}_l are the residuals for the levels l-1 and l respectively and $\mathbf{R}_0 = \mathbf{Y}$, the matrix of training image patches. This implies that the residual matrix in level l-1 serves as the training data for level l. Note that the sparsity of the representation in each level is fixed at 1. Hence, the overall approximation for all levels is

$$\mathbf{Y} = \sum_{l=1}^{L} \mathbf{\Psi}_l \mathbf{A}_l + \mathbf{R}_L. \tag{15}$$

MLD learning can be interpreted as a block-based dictionary learning problem with unit sparsity per block, where the sub-dictionary in each block can allow only a 1-sparse representation and each block corresponds to a level. The sub-dictionary for level l, Ψ_l , is the set of cluster centroids learned from the training matrix for that level, \mathbf{R}_{l-1} , using K-hyperline clustering. MLD learning can be formally stated as an optimization problem that proceeds from the first level until the stopping criteria is reached. For level l, we solve

$$\underset{\boldsymbol{\Psi}_{l}, \mathbf{A}_{l}}{\operatorname{argmin}} \|\mathbf{R}_{l-1} - \boldsymbol{\Psi}_{l} \mathbf{A}_{l}\|_{F}^{2} \text{ subject to } \|\mathbf{a}_{l,i}\|_{0} \leq 1,$$
 for $i = \{1, ..., T\},$ (16)

along with the constraint that the columns of Ψ_l have unit ℓ_2 norm, where $\mathbf{a}_{l,i}$ is the i^{th} column of \mathbf{A}_l and T is the

number of columns in \mathbf{A}_l . We adopt the notation $\{\Psi_l, \mathbf{A}_l\} = \mathrm{KLC}(\mathbf{R}_{l-1}, K_l)$ to denote the problem in (16) where K_l is the number of atoms in Ψ_l . The stopping criteria is provided either by imposing a limit on the residual representation error or the maximum number of levels (L). Note that the total number of levels is the same as the maximum number of non-zero coefficients (sparsity) of the representation. The error constraint can be stated as, $\|\mathbf{r}_{l,i}\|_2^2 \leq \epsilon, \forall i=1,...,T$, where $\mathbf{r}_{l,i}$ is the i^{th} column in \mathbf{R}_l , and ϵ is the error goal.

Table I lists the MLD learning algorithm with a fixed L. We use the notation $\Lambda_l(j)$ to denote the j^{th} element of the set Λ_l . The set Λ_l contains the indices of the residual vectors of level l whose norm is greater than the error goal. The residual vectors indexed by Λ_l are stacked in the matrix, $\hat{\mathbf{R}}_l$, which in turn serves as the training matrix for the next level, l+1. In MLD learning, for a given level l, the residual $\mathbf{r}_{l,i}$ is orthogonal to the representation $\Psi_l \mathbf{a}_{l,i}$. This implies that

$$\|\mathbf{r}_{l-1,i}\|_2^2 = \|\mathbf{\Psi}_l \mathbf{a}_{l,i}\|_2^2 + \|\mathbf{r}_{l,i}\|_2^2.$$
 (17)

Combining this with the fact that $\mathbf{y}_i = \sum_{l=1}^L \mathbf{\Psi}_l \mathbf{a}_{l,i} + \mathbf{r}_{L,i}$, $\mathbf{a}_{l,i}$ is 1-sparse, and the columns of $\mathbf{\Psi}_l$ are of unit ℓ_2 norm, we obtain the relation

$$\|\mathbf{y}_i\|_2^2 = \sum_{l=1}^L \|\mathbf{a}_{l,i}\|_2^2 + \|\mathbf{r}_{L,i}\|_2^2.$$
 (18)

Equation (18) states that the energy of any training vector is equal to the sum of squares of its coefficients and the energy of its residual. From (17), we also have that,

$$\|\mathbf{R}_{l-1}\|_F^2 = \|\mathbf{\Psi}_l \mathbf{A}_l\|_F^2 + \|\mathbf{R}_l\|_F^2. \tag{19}$$

The training vectors for the first level of the algorithm, $\mathbf{r}_{0,i}$ lie in the ambient \mathbb{R}^M space and the residuals, $\mathbf{r}_{1,i}$, lie in a finite union of \mathbb{R}^{M-1} subspaces. This is because, for each dictionary atom in the first level, its residual lies in an M-1 dimensional space orthogonal to it. In the second level, the dictionary atoms can possibly lie anywhere in \mathbb{R}^M , and hence the residuals can lie in a finite union of \mathbb{R}^{M-1} and \mathbb{R}^{M-2} dimensional subspaces. Hence, we can generalize that the dictionary atoms for all levels lie in \mathbb{R}^M , whereas the training vectors of level $l \geq 2$, lie in finite unions of $\mathbb{R}^{M-1}, \ldots, \mathbb{R}^{M-l+1}$ dimensional subspaces of the \mathbb{R}^M space.

B. Convergence

The convergence of MLD learning and the energy hierarchy in the representation obtained using an MLD can be shown by providing two guarantees. The first guarantee is that for a fixed number of atoms per level, the algorithm will converge to the required error within a sufficient number of levels. This is because the K-hyperline clustering makes the residual energy of the representation smaller than the energy of the training matrix at each level (i.e.) $\|\mathbf{R}_l\|_F^2 < \|\mathbf{R}_{l-1}\|_F^2$. This follows from (19) and the fact that $\|\mathbf{\Psi}_l\mathbf{A}_l\|_F^2 > 0$.

The second guarantee is that for a sufficient number of atoms per level, the representation energy in level l will be less than the representation energy in level l-1. To show this, we first state that for a sufficient number of dictionary

atoms per level, $\|\Psi_l \mathbf{A}_l\|_F^2 > \|\mathbf{R}_l\|_F^2$. This means that for every l

$$\|\mathbf{R}_l\|_F^2 < \|\mathbf{\Psi}_l \mathbf{A}_l\|_F^2 < \|\mathbf{R}_{l-1}\|_F^2,$$
 (20)

because of (19). This implies that $\|\mathbf{\Psi}_{l}\mathbf{A}_{l}\|_{F}^{2} < \|\mathbf{\Psi}_{l-1}\mathbf{A}_{l-1}\|_{F}^{2}$, i.e., the energy of the representation in each level reduces progressively from l=1 to l=L, thereby exhibiting energy hierarchy.

C. Estimating Number of Atoms in Each Level

The number of atoms in each level of an MLD can be optimally estimated using an information theoretic criteria such as minimum description length (MDL) [31]. The broad idea is that the model order, which is the number of dictionary atoms here, is chosen to minimize the total description length needed for representing the model and the data given the model. The codelength for encoding the data \mathbf{Y} given the model $\mathbf{\Theta}$ is given as the negative log likelihood $-\log p(\mathbf{Y}|\mathbf{\Theta})$. The description length for the model is the number of bits needed to code the model parameters.

In order to estimate the number of atoms in each level using the MDL principle, we need to make some assumptions on the residual obtained in each level. Our first assumption will be that the a fraction α of the total energy in each level E_l will be represented at that level and the remaining energy $(1-\alpha)E_l$ will be the residual energy. The residual and the representation energy sum up to the total energy in each level because, the residual in any level of MLD is orthogonal to the representation in that level. Therefore, at any level l, the represented energy will be $\alpha(1-\alpha)^{l-1}E$ and the residual energy will be $(1-\alpha)^{l}E$, where E is the total energy of training data at the first level. For simplicity, we also assume that the residual at each level follows the zero-mean multinormal distribution $\mathcal{N}(\mathbf{0}, \sigma_l^2 \mathbf{I}_M)$. Combining these two assumptions, the variance is estimated as $\sigma_l^2 = \frac{1}{MT}(1-\alpha)^l E$.

The total MDL score, which is an indicator of the information-theoretic complexity, is the sum of the negative log likelihood and the number of bits needed to encode the model. Encoding the model includes encoding the nonzero coefficients, their location, and the dictionary elements themselves. The MDL score for level l with the data \mathbf{R}_{l-1} , dictionary $\mathbf{\Psi}_l \in \mathbb{R}^{M \times K_l}$, and the coefficient matrix \mathbf{A}_l is

$$MDL(\mathbf{R}_{l-1}|\mathbf{\Psi}_{l}, \mathbf{A}_{l}, K_{l}) = \frac{1}{2\sigma_{l}^{2}} \sum_{i=1}^{T} \|\mathbf{r}_{l-1,i} - \mathbf{\Psi}_{l} \mathbf{a}_{l,i}\|_{2}^{2} + \frac{1}{2} T \log(MT) + T \log(TK_{l}) + \frac{1}{2} K_{l} M \log(MT).$$
 (21)

Here, the first term in the sum represents the data description length, which is also the negative log-likelihood of the data after ignoring the constant term. The second term is the number of bits needed to code the T non-zero coefficients as reals where each coefficient is coded using $0.5 \log(MT)$ bits [45]. The third term denotes the bits needed to code their locations which are integers between 1 and TK_l , and the fourth term represents the total bits needed to code all the dictionary elements as reals. The optimal model order K_l is the number of dictionary atoms that results in the least MDL

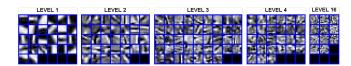


Fig. 1. The top 4 levels and the last level of the MLD dictionary where the number of atoms are estimated using the MDL procedure. It comprises of geometric patterns in the first few levels and stochastic textures in the last level. Since each level has a different number of atoms, each sub-dictionary is padded with zero vectors, which appear as black patches.

score. In practice, we test a finite number of model orders and pick the one which results in the least score. As an example, we train a dictionary using 5000 grayscale patches of size 8×8 from the BSDS dataset [46]. We preprocess the patches by vectorizing them and subtracting the mean of each vectorized patch from its elements. We perform MLD learning and estimate the estimate the optimal number of dictionary atoms in each level using $\alpha = 0.25$, for a maximum of 16 levels. For the sub-dictionary in each level, the number of atoms were varied between 10 and 50, and one that provided the least MDL score was chosen as optimal. The first few levels and the last level of the MLD obtained using such procedure is shown in Figure 1. The minimum MDL score obtained in each level is shown in 2. From these two figures, clearly, the informationtheoretic complexity of the sub-dictionaries increase with the number of levels, and the atoms themselves progress from being simple geometric structures to stochastic textures.

D. Sparse Approximation using an MLD

In order to compute sparse codes for novel test data using a multilevel dictionary, we propose to perform reconstruction using a Multilevel Pursuit (MulP) procedure which evaluates a 1-sparse representation for each level using the dictionary atoms from that level. Therefore, the coefficient vector for the i^{th} data sample $\mathbf{r}_{l,i}$ in level l is obtained using a simple correlate-andmax operation, whereby we compute the correlations $\mathbf{\Psi}_l^T\mathbf{r}_{l,i}$ and pick the coefficient value and index corresponding to the maximum absolute correlation. The computational complexity of a correlate-and-max operation is of order MK_l and hence the complexity of obtaining the full representation using L levels is of order MK, where $K = \sum_{i=1}^L K_l$ is the total number of atoms in the dictionary. Whereas, the complexity of obtaining an L sparse representation on the full dictionary using Orthogonal Matching Pursuit is of order LMK.

E. Robust Multilevel Dictionaries

Although MLD learning is a simple procedure capable of handling large scale data with useful asymptotic generalization properties as described in Section (IV-D), the procedure can be made robust and its generalization performance can be improved using randomization schemes. The Robust MLD (RMLD) learning scheme, which is closely related to *Rvotes* [47] - a supervised ensemble learning method, improves the generalization performance of MLD as evidenced by Figure 8. The *Rvotes* scheme randomly samples the training set to create D sets of T_D samples each, where $T_D \ll T$. The final prediction is obtained by averaging the predictions from

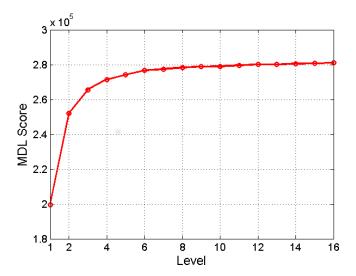


Fig. 2. The minimum MDL score of each level. The information-theoretic complexity of the sub-dictionaries increase with the number of levels.

the multiple hypotheses learned from the training sets. For learning level l in RMLD, we draw D subsets of randomly chosen training samples, $\{\mathbf{Y}_l^{(d)}\}_{d=1}^D$ from the original training set \mathbf{Y}_l of size T, allowing for overlap across the subsets. Note that here, $\mathbf{Y}_l = \mathbf{R}_{l-1}$. The superscript here denotes the index of the subset. For each subset $\mathbf{Y}_l^{(d)}$ of size $T_D \ll T$, we learn a sub-dictionary $\mathbf{\Psi}_l^{(d)}$ with K_l atoms using K-hyperline clustering. For each training sample in \mathbf{Y}_l , we compute 1-sparse representations using all the D sub-dictionaries, and denote the set of coefficient matrices as $\{\mathbf{A}_l^{(d)}\}_{d=1}^D$. The approximation for the i^{th} training sample in level l, $\mathbf{y}_{l,i}$, is computed as the average of approximations using all D sub-dictionaries, $\frac{1}{D}\sum_d\mathbf{\Psi}_l^{(d)}\mathbf{a}_{l,i}^{(d)}$. The ensemble approximations for all training samples in the level can be used to compute the set of residuals, and this process is repeated for a desired number of levels, to obtain an RMLD.

Reconstruction of test data with an RMLD is performed by extending the multilevel pursuit. We obtain D approximations for each data sample at a given level, average the approximations, compute the residual and repeat this for the subsequent levels. Note that this can also be implemented as multiple *correlate-and-max* operations per data sample per level. Clearly, the computational complexity for obtaining a sparse representation using the RMLD is of order DMK, where $K = \sum_{i=1}^{L} K_i$.

IV. STABILITY AND GENERALIZATION

In this section, the behavior of the proposed dictionary learning algorithm is considered from the viewpoint of algorithmic stability: the behavior of the algorithm with respect to the perturbations in the training set. It will be shown that the dictionary atoms learned by the algorithm from two different training sets whose samples are realized from the same probability space, become arbitrarily close to each other, as the number of training samples $T \to \infty$. Since the proposed MLD learning is equivalent to learning K-hyperline cluster centroids in multiple levels, the stability analysis of

K-hyperline clustering [38], briefly discussed in Section II-B, will be utilized in order to prove its stability. For each level of learning, the cases of single and multiple minimizers to the clustering objective will be considered. Proving that the learning algorithm is stable will show that the global dictionaries learned from the data depend only on the probability space to which the training samples belong and not on the actual samples themselves, as $T \to \infty$. We also show that the MLD learning generalizes asymptotically, i.e., the difference between expected error and average empirical error in training approaches zero, as $T \to \infty$. Therefore, the expected error for novel test data, drawn from the same distribution as the training data, will approach the average empirical training error.

The stability analysis of the MLD algorithm will be performed by considering two different dictionaries Ψ and Λ with L levels each. Each level consists of K_l dictionary atoms and the sub-dictionaries in each level are indicated by Ψ_l and Λ_l respectively. Sub-dictionaries Ψ_l and Λ_l are the cluster centers learned using K-hyperline clustering on the training data for level l. The steps involved in proving the overall stability of the algorithm are: (a) showing that each level of the algorithm is stable in terms of $L_1(P)$ distance between the distortion functions, defined in (10), as the number of training samples $T \to \infty$ (Section IV-A), (b) proving that stability in terms of $L_1(P)$ distances indicates closeness of the centers of the two clusterings (Section IV-B), in terms of the metric defined in (12), and (c) showing that level-wise stability leads to overall stability of the dictionary learning algorithm (Section IV-C).

A. Level-wise Stability

Let us define a probability space $(\mathcal{Y}_l, \mathbf{\Sigma}_l, P_l)$ where \mathcal{Y}_l is the data that lies in \mathbb{R}^M , and P_l is the probability measure. The training samples for the sub-dictionaries $\mathbf{\Psi}_l$ and $\mathbf{\Lambda}_l$ are two different sets of T i.i.d. realizations from the probability space. We also assume that the ℓ_2 norm of the training samples is bounded from above and below (i.e.), $0 < r \le ||\mathbf{y}||_2 \le R < \infty$. Note that, in a general case, the data will lie in \mathbb{R}^M for the first level of dictionary learning and in a finite union of lower-dimensional subspaces of \mathbb{R}^M for the subsequent levels. In both cases, the following argument on stability will hold. This is because when the training data lies in a union of lower dimensional subspaces of \mathbb{R}^M , we can assume it to be still lying in \mathbb{R}^M , but assign the probabilities outside the union of subspaces to be zero.

The distortion function class for the clusterings, defined similar to (6), is uniform Donsker because the covering number with respect to the supremum norm grows polynomially, according to (9). When a unique minimizer exists for the clustering objective, the distortion functions corresponding to the different clusterings Ψ_l and Λ_l become arbitrarily close, $\|g_{\Psi_l} - g_{\Lambda_l}\|_{L_1(P_l)} \stackrel{P}{\to} 0$, even for completely disjoint training sets, as $T \to \infty$. However, in the case of multiple minimizers, $\|g_{\Psi_l} - g_{\Lambda_l}\|_{L_1(P_l)} \stackrel{P}{\to} 0$ holds only with respect to a change of $o(\sqrt{T})$ training samples between the two clusterings, and fails to hold for a change of $\Omega(\sqrt{T})$ samples [35], [38].

B. Distance between Cluster Centers for a Stable Clustering

For each cluster center in the clustering Ψ_l , we pick the closest cluster center from Λ_l , in terms of the distortion measure (4), and form pairs. Let us indicate the $j^{\rm th}$ pair of cluster centers by $\psi_{l,j}$ and $\lambda_{l,j}$. Let us define τ disjoint sets $\{A_i\}_{i=1}^{\tau}$, in which the training data for the clusterings exist, such that $P_l(\cup_{i=1}^{\tau} A_i) = 1$. By defining such disjoint sets, we can formalize the notion of training data lying in a union of subspaces of \mathbb{R}^M . The intuitive fact that the cluster centers of two clusterings are close to each other, given that their distortion functions are close, is proved in the lemma below.

Lemma 4.1: Consider two sub-dictionaries (clusterings) Ψ_l and Λ_l with K_l atoms each obtained using the T training samples that exist in the τ disjoint sets $\{A_i\}_{i=1}^{\tau}$ in the \mathbb{R}^M space, with $0 < r \leq \|\mathbf{y}\|_2 \leq R < \infty$, and $dP_l(\mathbf{y})/d\mathbf{y} > C$ in each of the sets. When the distortion functions become arbitrarily close to each other, $\|g_{\Psi_l} - g_{\Lambda_l}\|_{L_1(P_l)} \xrightarrow{P} 0$ as $T \to \infty$, the smallest angle between the subspaces spanned by the cluster centers becomes arbitrarily close to zero, i.e.,

$$\angle(\boldsymbol{\psi}_{l,j}, \boldsymbol{\lambda}_{l,j}) \xrightarrow{P} 0, \forall j \in 1, \dots, K_l.$$
 (22)

Proof: Denote the smallest angle between the subspaces represented by $\psi_{l,j}$ and $\lambda_{l,j}$ as $\angle(\psi_{l,j},\lambda_{l,j}) = \rho_{l,j}$ and define a region $S(\psi_{l,j},\rho_{l,j}/2) = \{\mathbf{y}|\angle(\psi_{l,j},\mathbf{y}) \leq \rho_{l,j}/2, 0 < r \leq \|\mathbf{y}\|_2 \leq R < \infty\}$. If $\mathbf{y} \in S(\psi_{l,j},\rho_{l,j}/2)$, then $\mathbf{y}^T(\mathbf{I}-\psi_{l,j}\psi_{l,j}^T)\mathbf{y} \leq \mathbf{y}^T(\mathbf{I}-\lambda_{l,j}\lambda_{l,j}^T)\mathbf{y}$. An illustration of this setup for a 2-D case is given in Figure 3. In this figure, the arc $\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2$ is of radius r and represents the minimum value of $\|\mathbf{y}\|_2$. By definition, the $L_1(P_l)$ distance between the distortion functions of the clusterings for data that exists in the disjoint sets $\{A_i\}_{i=1}^T$ is

$$\|g_{\mathbf{\Psi}_l} - g_{\mathbf{\Lambda}_l}\|_{L_1(P_l)} = \sum_{i=1}^{\tau} \int_{A_i} |g_{\mathbf{\Psi}_l}(\mathbf{y}) - g_{\mathbf{\Lambda}_l}(\mathbf{y})| dP_l(\mathbf{y}). \quad (23)$$

For any j and i with a non-empty $B_{l,i,j} = S(\psi_{l,j}, \rho_{l,j}/2) \cap A_i$ we have,

$$\|g_{\mathbf{\Psi}_l} - g_{\mathbf{\Lambda}_l}\|_{L_1(P_l)} \ge \int_{B_{l,i,j}} |g_{\mathbf{\Psi}_l}(\mathbf{y}) - g_{\mathbf{\Lambda}_l}(\mathbf{y})| dP_l(\mathbf{y}), \quad (24)$$

$$= \int_{B_{l,i,j}} \left[\mathbf{y}^{T} \left(\mathbf{I} - \boldsymbol{\lambda}_{l,j} \boldsymbol{\lambda}_{l,j}^{T} \right) \mathbf{y} - \sum_{k=1}^{K} \mathbf{y}^{T} \left(\mathbf{I} - \boldsymbol{\psi}_{l,k} \boldsymbol{\psi}_{l,k}^{T} \right) \mathbf{y} \right]$$

$$\mathbb{I} \left(\mathbf{y} \text{ closest to } \boldsymbol{\psi}_{l,k} \right) dP_{l}(\mathbf{y}), \tag{25}$$

$$\geq \int_{B_{l,i,j}} \left[\mathbf{y}^T \left(\mathbf{I} - \boldsymbol{\lambda}_{l,j} \boldsymbol{\lambda}_{l,j}^T \right) \mathbf{y} - \mathbf{y}^T \left(\mathbf{I} - \boldsymbol{\psi}_{l,j} \boldsymbol{\psi}_{l,j}^T \right) \mathbf{y} \right] dP_l(\mathbf{y}), \tag{26}$$

$$\geq C \int_{B_{l,i,j}} \left[\left(\mathbf{y}^T \boldsymbol{\psi}_{l,j} \right)^2 - \left(\mathbf{y}^T \boldsymbol{\lambda}_{l,j} \right)^2 \right] d\mathbf{y}. \tag{27}$$

We have $g_{\mathbf{\Lambda}_l}(\mathbf{y}) = \mathbf{y}^T \left(\mathbf{I} - \mathbf{\lambda}_{l,j} \mathbf{\lambda}_{l,j}^T \right) \mathbf{y}$ in (25), since $\mathbf{\lambda}_{l,j}$ is the closest cluster center to the data in $S(\psi_{l,j}, \rho_{l,j}/2) \cap A_i$ in terms of the distortion measure (4). Note that \mathbb{I} is the indicator function and (27) follows from (26) because $dP_l(\mathbf{y})/d\mathbf{y} > C$. Since by assumption, $\|g_{\mathbf{\Psi}_l} - g_{\mathbf{\Lambda}_l}\|_{L_1(P_l)} \xrightarrow{P} 0$, from (27), we have

$$(\mathbf{y}^T \boldsymbol{\psi}_{l,j})^2 - (\mathbf{y}^T \boldsymbol{\lambda}_{l,j})^2 \xrightarrow{P} 0,$$
 (28)

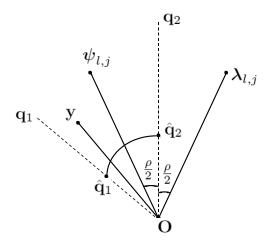


Fig. 3. Illustration for showing the stability of cluster centroids from the stability of distortion function.

because the integrand in (27) is a continuous non-negative function in the region of integration.

Denoting the smallest angles between \mathbf{y} and the subspaces spanned by $\psi_{l,j}$ and $\lambda_{l,j}$ to be $\theta_{\psi_{l,j}}$ and $\theta_{\lambda_{l,j}}$ respectively, from (28), we have $\|\mathbf{y}\|_2^2(\cos^2\theta_{\psi_{l,j}}-\cos^2\theta_{\lambda_{l,j}})\overset{P}{\to}0$, for all \mathbf{y} . By definition of the region $B_{l,i,j}$, we have $\theta_{\psi_{l,j}} \leq \theta_{\lambda_{l,j}}$. Since $\|\mathbf{y}\|_2$ is bounded away from zero and infinity, if $(\cos^2\theta_{\psi_{l,j}}-\cos^2\theta_{\lambda_{l,j}})\overset{P}{\to}0$ holds for all $\mathbf{y}\in B_{l,i,j}$, then we have $\angle(\psi_{l,j},\lambda_{l,j})\overset{P}{\to}0$. This is true for all cluster center pairs as we have shown this for an arbitrary i and j.

C. Stability of the MLD Algorithm

The stability of the MLD algorithm as a whole, is proved in Theorem 4.3 from its level-wise stability by using an induction argument. The proof will depend on the following lemma which shows that the residuals from two stable clusterings belong to the same probability space.

Lemma 4.2: When the training vectors for the subdictionaries (clusterings) Ψ_l and Λ_l are obtained from the probability space $(\mathcal{Y}_l, \Sigma_l, P_l)$, and the cluster center pairs become arbitrarily close to each other as $T \to \infty$, the residual vectors from both the clusterings belong to an identical probability space $(\mathcal{Y}_{l+1}, \Sigma_{l+1}, P_{l+1})$.

Proof: For the j^{th} cluster center pair $\psi_{l,j}$, $\lambda_{l,j}$, define $\bar{\Psi}_{l,j}$ and $\bar{\Lambda}_{l,j}$ as the projection matrices for their respective orthogonal complement subspaces $\psi_{l,j}^{\perp}$ and $\lambda_{l,j}^{\perp}$. Define the sets $D_{\psi_{l,j}} = \{ \mathbf{y} \in \bar{\Psi}_{l,j}(\beta + d\beta) + \psi_{l,j}\alpha \}$ and $D_{\lambda_{l,j}} = \{ \mathbf{y} \in \bar{\Lambda}_{l,j}(\beta + d\beta) + \lambda_{l,j}\alpha \}$, where $-\infty < \alpha < \infty$, β is an arbitrary fixed vector, not orthogonal to both $\psi_{l,j}$ and $\lambda_{l,j}$, and $d\beta$ is a differential element. The residual vector set for the cluster $\psi_{l,j}$, when $\mathbf{y} \in D_{\psi_{l,j}}$ is given by, $\mathbf{r}_{\psi_{l,j}} \in \{\bar{\Psi}_{l,j}(\beta + d\beta)\}$. Similarly for the cluster $\lambda_{l,j}$, we have $\mathbf{r}_{\lambda_{l,j}} \in \{\bar{\Lambda}_{l,j}(\beta + d\beta)\}$. For a 2-D case, Figure 4 shows the 1-D subspace $\psi_{l,j}$, its orthogonal complement $\psi_{l,j}^{\perp}$, the set $D_{\psi_{l,j}}$ and the residual set $\{\bar{\Psi}_{l,j}(\beta + d\beta)\}$.

In terms of probabilities, we also have that $P_l(\mathbf{y} \in D_{\psi_{l,j}}) = P_{l+1}(\mathbf{r}_{\psi_{l,j}} \in \{\bar{\Psi}_{l,j}(\boldsymbol{\beta} + d\boldsymbol{\beta})\})$, because the residual set

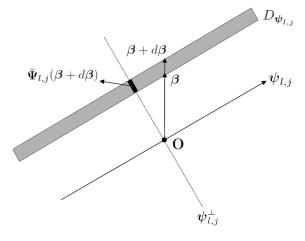


Fig. 4. The residual set $\{\bar{\Psi}_{l,j}(\beta+d\beta)\}$, for the 1-D subspace $\psi_{l,j}$, lying in its orthogonal complement subspace $\psi_{l,j}^{\perp}$.

 $\{\bar{\Psi}_{l,j}(\beta+d\beta)\}$ is obtained by a linear transformation of $D_{\psi_{l,j}}$. Here P_l and P_{l+1} are probability measures defined on the training data for levels l and l+1 respectively. Similarly, $P_l(\mathbf{y}\in D_{\lambda_{l,j}})=P_{l+1}(\mathbf{r}_{\lambda_{l,j}}\in\{\bar{\Lambda}_{l,j}(\beta+d\beta)\})$. When $T\to\infty$, the cluster center pairs become arbitrarily close to each other, i.e., $\angle(\psi_{l,j},\lambda_{l,j})\stackrel{P}{\to}0$, by assumption. Therefore, the symmetric difference between the sets $D_{\psi_{l,j}}$ and $D_{\lambda_{l,j}}$ approaches the null set, which implies that $P_l(\mathbf{y}\in D_{\psi_{l,j}})-P_l(\mathbf{y}\in D_{\lambda_{l,j}})\to 0$. This implies,

$$P_{l+1}(\mathbf{r}_{\boldsymbol{\psi}_{l,j}} \in \{\bar{\boldsymbol{\Psi}}_{l,j}(\boldsymbol{\beta} + d\boldsymbol{\beta})\}) - P_{l+1}(\mathbf{r}_{\boldsymbol{\lambda}_{l,j}} \in \{\bar{\boldsymbol{\Lambda}}_{l,j}(\boldsymbol{\beta} + d\boldsymbol{\beta})\}) \to 0,$$
 (29)

for an arbitrary β and $d\beta$, as $T \to \infty$. This means that the residuals of $\psi_{l,j}$ and $\lambda_{l,j}$ belong to a unique but identical probability space. Since we proved this for an arbitrary l and j, we can say that the residuals of clusterings Ψ_l and Λ_l belong to an identical probability space given by $(\mathcal{Y}_{l+1}, \Sigma_{l+1}, P_{l+1})$.

Theorem 4.3: Given that the training vectors for the first level are generated from the probability space $(\mathcal{Y}_1, \Sigma_1, P_1)$, and the norms of training vectors for each level are bounded as $0 < r \le ||\mathbf{y}||_2 \le R < \infty$, the MLD learning algorithm is stable as a whole.

Proof: The level-wise stability of MLD was shown in Section IV-A, for two cases: (a) when a unique minimizer exists for the distortion function and (b) when a unique minimizer does not exist. Lemma 4.1 proved that the stability in terms of closeness of distortion functions implied stability in terms of learned cluster centers. For showing the level-wise stability, we assumed that the training vectors in level l for clusterings Ψ_l and Λ_l belonged to the same probability space. However, when learning the dictionary, this is true only for the first level, as we supply the algorithm with training vectors from the probability space $(\mathcal{Y}_1, \Sigma_1, P_1)$.

We note that the training vectors for level l+1 are residuals of the clusterings Ψ_l and Λ_l . Lemma 4.2 showed that the residuals of level l for both the clusterings belong to an identical probability space $(\mathcal{Y}_{l+1}, \Sigma_{l+1}, P_{l+1})$, given that the training vectors of level l are realizations from the probability

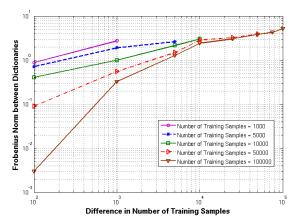


Fig. 5. Demonstration of the stability behavior of the proposed MLD learning algorithm. The minimum Frobenius norm between difference of two dictionaries with respect to permutation of their columns and signs is shown. The second dictionary is obtained by replacing different number of samples in the training set, used for training the original dictionary, with new data samples.

space $(\mathcal{Y}_l, \Sigma_l, P_l)$ and $T \to \infty$. By induction, this along with the fact that the training vectors for level 1 belong to the same probability space $(\mathcal{Y}_1, \Sigma_1, P_1)$, shows that all the training vectors of both the dictionaries for any level l indeed belong to a probability space $(\mathcal{Y}_l, \Sigma_l, P_l)$ corresponding to that level. Hence all the levels of the dictionary learning are stable and the MLD learning is stable as a whole. Similar to K-hyperline clustering, if there are multiple minimizers in at least one level, the algorithm is stable only with respect to a change of $o(\sqrt{T})$ training samples between the two clusterings and failts to hold for a change of $\Omega(\sqrt{T})$ samples.

D. Generalization Analysis

Since our learning algorithm consists of multiple levels, and cannot be expressed as an ERM on a whole, the algorithm can be said to generalize asymptotically if the sum of empirical errors for all levels converge to the sum of expected errors, as $T \to \infty$. This can be expressed as

$$\left| \frac{1}{T} \sum_{l=1}^{L} \sum_{i=1}^{T} g_{\Psi_l}(\mathbf{y}_{l,i}) - \sum_{l=1}^{L} \mathbb{E}_{P_l}[g_{\Psi_l}] \right| \xrightarrow{P} 0, \quad (30)$$

where the training samples for level l given by $\{\mathbf{y}_{l,i}\}_{i=1}^{T}$ are obtained from the probability space $(\mathcal{Y}_l, \mathbf{\Sigma}_l, P_l)$. When (30) holds and the learning algorithm generalizes, it can be seen that the expected error for test data which is drawn from the same probability space as that of the training data, is close to the average empirical error. Therefore, when the cluster centers for each level are obtained by minimizing the empirical error, the expected test error will also be small.

In order to show that (30) holds, we use the fact that each level of MLD learning is obtained using K-hyperline clustering. Hence, from (8), the average empirical distortion in each level converges to the expected distortion as $T \to \infty$,

$$\left| \frac{1}{T} \sum_{i=1}^{T} g_{\mathbf{\Psi}_l}(\mathbf{y}_{l,i}) - \mathbb{E}_{P_l}[g_{\mathbf{\Psi}_l}] \right| \xrightarrow{P} 0. \tag{31}$$

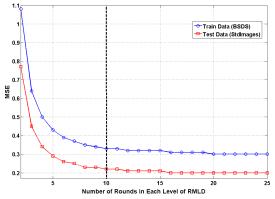


Fig. 6. Choosing the number of rounds (R) in RMLD learning. In this demonstration, RMLD design was carried out using 100,000 samples and we observed that beyond 10, both the train MSE and the test MSE do not change significantly.

The validity of the condition in (30) follows directly from the triangle inequality,

$$\left| \frac{1}{T} \sum_{l=1}^{L} \sum_{i=1}^{T} g_{\mathbf{\Psi}_{l}}(\mathbf{y}_{l,i}) - \sum_{l=1}^{L} \mathbb{E}_{P_{l}}[g_{\mathbf{\Psi}_{l}}] \right|$$

$$\leq \sum_{l=1}^{L} \left| \frac{1}{T} \sum_{i=1}^{T} g_{\mathbf{\Psi}_{l}}(\mathbf{y}_{l,i}) - \mathbb{E}_{P_{l}}[g_{\mathbf{\Psi}_{l}}] \right|. \tag{32}$$

If the MulP coding scheme is used for test data, and the training and test data for level 1 are obtained from the probability space $(\mathcal{Y}_1, \Sigma_1, P_1)$, the probability space for both training and test data in level l will be $(\mathcal{Y}_l, \Sigma_l, P_l)$. This is because, both the MulP coding scheme and MLD learning associate the data to a dictionary atom using the maximum absolute correlation measure and create a residual that is orthogonal to the atom chosen in a level. Hence, the assumption that training and test data are drawn from the same probability space in all levels hold and the expected test error will be similar to the average empirical training error.

V. SIMULATION RESULTS

In this section, we present experiments to demonstrate the stability and generalization characteristics of a multilevel dictionary, and evaluate its use in compressed recovery of images and subspace learning. Both stability and generalization are crucial for building effective global dictionaries that can model patterns in any novel test image. Although it is not possible to demonstrate the asymptotic behavior experimentally, we study the changes in the behavior of the learning algorithm with increase in the number of samples used for training. Compressed recovery is a highly relevant application for global dictionaries, since it is not possible to infer dictionaries with good reconstructive power directly from the low-dimensional random measurements of image patches. It is typical to employ both ℓ_1 minimization and greedy pursuit methods for recovering images from their compressed measurements. Though ℓ_1 minimization incurs higher computational complexity, it often provides improved recovery performance when compared to greedy approaches. Hence, it is important to compare its recovery performance to that of the MLD that uses a simple

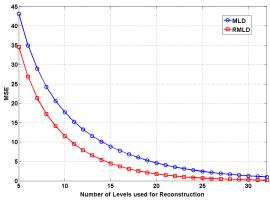


Fig. 7. Reconstruction of novel test data using MLD and RMLD dictionaries for the case T=100,000. The approximation error is plotted against the number of levels used for reconstruction with both the dictionaries.

greedy pursuit. Subspace learning is another application that can benefit from the use of multilevel dictionaries. In subspace learning, it is common to obtain a linear embedding from the training data, and apply it to novel test data for dimensionality reduction, classification, and visualization. These approaches can be unsupervised (eg. Principal Component Analysis, Locality Preserving Projections) or can use the class label information while learning the embedding (eg. Linear Discriminant Analysis, Local Discriminant Embedding). Several subspace learning algorithms can be unified under the framework of graph embedding [48], wherein an undirected graph describing the relation between the data samples is provided as the input. We propose to use graphs constructed based on sparse codes, from a multilevel dictionary, for subspace learning in both supervised and unsupervised settings.

All simulations for stability/generalization, and compressed recovery use dictionaries trained on image patches from the Berkeley Segmentation Dataset (BSDS) [46]. The BSDS dataset contains a total of 400 images and the number of patches used in our experiments vary between 5000 and 400,000. The images were converted to grayscale and no other preprocessing was performed on these images. We used patches of size 8×8 and no noise was added to the patches. For evaluating the performance of the dictionaries, we considered 8 standard images (*Barbara, Boat, House, Lena, Couple, Finger-print, Man, Peppers*). For the subspace learning simulations, we used the Forest Covertype dataset [49] which consists of 581,012 samples belonging to 7 different classes. As per the standard procedure, we used the first 15,120 samples (2160 per class) for training and the rest for testing.

A. Stability

In order to illustrate the stability characteristics of MLD learning, we setup an experiment where we consider a multilevel dictionary of 4 levels, with 8 atoms in each level. We trained multilevel dictionaries using different number of training patches T. As we showed in Section IV, asymptotic stability is guaranteed when the training set is changed by not more than $o(\sqrt{T})$ samples. The inferred dictionary atoms will not vary significantly, if this condition is satisfied. We fixed the size of the training set at different values $T=\{1000,$

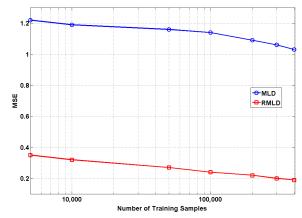


Fig. 8. Demonstration of the generalization characteristics of the proposed MLD and RMLD algorithms. We plot the MSE obtained by representing patches from the test dataset, using dictionaries learned with different number of training patches.

5000, 10,000, 50,000, 100,000} and learned an initial set of dictionaries using the proposed algorithm. The second set of dictionaries were obtained by replacing different number of samples from the original training set. For each case of T, the number of replaced samples was varied between 100 and T. For example, when T = 10,000, the number of replaced training samples were 100, 1000, 5000, and 10,000. The amount of change between the initial and the second set of dictionaries was quantified using the minimum Frobenius norm of their difference with respect to permutations of their columns and sign changes. In Figure 5, we plot this quantity for different values of T as a function of the number of samples replaced in the training set. For each case of T, the difference between the dictionaries increases as we increase the replaced number of training samples. Furthermore, for a fixed number of replaced samples (say 100), the difference reduces with the increase in the number of training samples, since it becomes closer to asymptotic behavior.

B. Generalization

Generalization of a dictionary learning algorithm guarantees a small approximation error for a test data sample, if the training samples are well approximated by the dictionary. In order to demonstrate the generalization characteristics of MLD learning, we designed dictionaries using different number of training image patches, of size 8×8 , and evaluated the sparse approximation error for patches in the test dataset. The test dataset consisted of 120,000 patches chosen randomly from the 8 standard images. For multilevel learning, we fixed the number of levels at 32, and used the approach proposed in Section III-C to estimate the number of atoms needed in each level ($\alpha = 0.5$). Similarly, we fixed the number of levels at 32 for the RMLD learning. Since RMLD learning does not require careful choice of the number of atoms in each level, we fixed $K_{\ell} = 32$. Though learning multiple sets of atoms in each level can lead to improved generalization, the benefit seems to level off after a certain number of rounds. As an example, let us consider the case where T = 100,000 and vary the number of rounds in RMLD between 2 and 25. As described in Section III-E, increasing the number of rounds results in

higher computational complexity while evaluating the sparse codes. Figure 6 illustrates the MSE on the training data and the test data obtained using RMLD with different number of rounds in each level. Since the MSE did not vary significantly beyond 10 rounds, we fixed R=10 in our reconstruction experiments.

Figure 7 compares the approximation error (MSE) obtained for the test dataset with MLD and RMLD (10 rounds) respectively, for the case of T=100,000. The figure plots the MSE against the number of levels used in the reconstruction algorithm. Figure 8 shows the approximation error (MSE) for the test image patches obtained with MLD and RMLD dictionaries learned using different number of training samples (varied between 5000 and 400,000). Since we proved in Section IV-D the MLD learning generalizes asymptotically, we expect the approximation error for the test data to reduce with the increase in the size of the training set. From both these figures, it is clear that the RMLD scheme results in improved approximation of novel test patches when compared to MLD.

C. Application: Compressed Recovery

In compressed recovery, an image is recovered using the low-dimensional random projections obtained from its patches. The performance of compressed recovery based on random measurement systems is compared for MLD, RMLD and online dictionaries. For the case of online dictionaries learned using the algorithm described in [19], we report results obtained using both ℓ_1 minimization and the OMP algorithm. Sensing and recovery were performed on a patch-by-patch basis, on non-overlapping patches of size 8×8 . The multilevel dictionaries were obtained with the parameters described in the previous section, using 400,000 training samples. The online dictionary was trained using the same training set, with the number of atoms fixed at 1024. The measurement process can described as $\mathbf{x} = \mathbf{\Phi} \mathbf{\Psi} \mathbf{a} + \boldsymbol{\eta}$ where $\mathbf{\Psi}$ is the dictionary, $\mathbf{\Phi}$ is the measurement or projection matrix, η is the AWGN vector added to the measurement process, x is the output of the measurement process, and a is the sparse coefficient vector such that $y = \Psi a$. The size of the data vector y is $M \times 1$, that of Ψ is $M \times K$, that of the measurement matrix Φ is $N \times M$, where N < M, and that of the measured vector x is $N \times 1$. The entries in the random measurement matrix were independent realizations from a standard normal distribution. We recover the underlying image from its compressed measurements, using online (OMP, ℓ_1), MLD, and RMLD dictionaries. For each case, we present average results from 100 trial runs, each time with a different measurement matrix. The recovery performance was evaluated for the set of standard images and reported in Table II. Figure 9 illustrates the recovered images obtained using different dictionaries with 8 random measurements under noise (SNR = 15 dB). We observed that the MulP reconstruction using the proposed MLD dictionary resulted in improved recovery performance, at different measurement conditions, when compared to using greedy pursuit (OMP) with the online dictionary. However, both the MulP reconstruction for RMLD and ℓ_1 -based reconstruction with the online dictionary perform significantly better than the other two approaches. In particular, the RMLD reconstruction achieves improved recovery at reduced number of measurements (8, 16) and in presence of noise.

D. Application: Subspace Learning

In this section, we evaluate the use of sparse codes obtained with multilevel dictionaries in unsupervised and supervised subspace learning. In particular, we use the locality preserving projections (LPP) [50] and local discriminant embedding (LDE) [51] approaches to perform classification on the Forest Covertype dataset. LPP is an unsupervised embedding approach which computes projection directions such that the pairwise distances of the projected training samples in the neighborhood are preserved. Let us define the training data as $\{\mathbf{y}_i|\mathbf{y}_i\in\mathbb{R}^M\}_{i=1}^T$. An undirected graph G is defined, with the training samples as vertices, and the similarity between the neighboring training samples are coded in the affinity matrix $\mathbf{W} \in \mathbb{R}^{T \times T}$. In the proposed setup, we learn a dictionary using the training samples and compute the affinity matrix $\mathbf{W} = |\mathbf{A}^T \mathbf{A}|$, where \mathbf{A} is the matrix of sparse coefficients. Following this, we sparsify W by retaining only the τ largest similarities for each sample. Note that this construction is different from the ℓ_1 graph construction in [16] and computationally efficient. Let us denote the graph Laplacian as L = D - W, where D is a degree matrix with each diagonal element containing the sum of the corresponding row or column of W. The d projection directions for LPP, $\mathbf{V} \in \mathbb{R}^{M \times d}$, where d < M, can be computed by optimizing

$$\min_{\text{trace}(\mathbf{V}^T\mathbf{Y}\mathbf{D}\mathbf{Y}^T\mathbf{V})=\mathbf{I}} \text{trace}(\mathbf{V}^T\mathbf{Y}\mathbf{L}\mathbf{Y}^T\mathbf{V}). \tag{33}$$

Here \mathbf{Y} is a matrix obtained by stacking all data samples as its columns. The embedding for any data sample \mathbf{z} can be obtained as $\mathbf{z} = \mathbf{V}^T \mathbf{y}$. In a supervised setting, we define the intra-class and inter-class affinity matrices, \mathbf{W} and \mathbf{W}' respectively, as

$$w_{ij} = \begin{cases} |\mathbf{a}_i^T \mathbf{a}_j| & \text{if } \pi_i = \pi_j \text{ AND } j \in \mathcal{N}_\tau(i), \\ 0 & \text{otherwise,} \end{cases}$$
(34)

$$w'_{ij} = \begin{cases} |\mathbf{a}_i^T \mathbf{a}_j| & \text{if } \pi_i \neq \pi_j \text{ AND } j \in \mathcal{N}_{\tau'}(i), \\ 0 & \text{otherwise,} \end{cases}$$
(35)

where π_i is the label of the i^{th} training sample, and $\mathcal{N}_{\tau}(i)$ and $\mathcal{N}_{\tau'}(i)$ are the sets that contain the indices of τ intraclass and τ' inter-class neighbors of the i^{th} training sample. The neighbors of a sample i are sorted based on the order of decreasing absolute correlations of their sparse code with \mathbf{a}_i . Using these affinity matrices, local discriminant embedding is performed by solving

$$\underset{\mathbf{V}}{\operatorname{argmax}} \frac{\operatorname{Tr}[\mathbf{V}^{T}\mathbf{X}^{T}\mathbf{L}'\mathbf{X}\mathbf{V}]}{\operatorname{Tr}[\mathbf{V}^{T}\mathbf{X}^{T}\mathbf{L}\mathbf{X}\mathbf{V}]}.$$
 (36)

For both the subspace learning approaches, we varied the number of training samples between 250 and 2160 per class and fixed the embedding dimension d=30. For MLD and RMLD learning, we fixed the number of levels at 32 and the number of rounds, R, for RMLD was fixed at 30. For

TABLE II

PSNR (dB) of the images recovered from compressed measurements obtained using Gaussian random measurement matrices. Results obtained with the Online (OMP), Online (ℓ_1), RMLD (MulP), and MLD (MulP) algorithms are given in clockwise order beginning from top left corner. Higher PSNR for each case is indicated in bold font.

SNR (dB)	# Measurements	Bo	oat	Но	use	Le	ena	M	an	Pep	pers
0	8	21.43	22.32	22.43	23.36	23.13	24.05	22.23	23.13	19.08	20.01
		21.60	22.43	22.86	23.76	23.52	24.39	22.43	23.28	19.55	20.43
	16	22.19	23.19	23.31	24.39	24.02	25.03	22.98	23.97	19.97	21.05
		22.67	23.60	24.18	25.15	24.76	25.75	23.51	24.46	20.96	21.95
	32	23.50	24.18	24.94	25.48	25.54	26.08	24.26	24.95	21.71	22.14
		24.18	25.15	25.94	27.03	26.46	27.54	25.02	26.02	22.87	23.90
15	8	22.77	23.65	23.95	24.97	24.61	25.60	23.58	24.46	20.61	21.68
		23.48	24.45	25.11	26.14	25.69	26.70	24.34	25.31	21.90	22.90
	16	23.94	26.33	25.36	28.65	26.03	28.92	24.74	27.08	22.28	25.43
		25.29	26.56	27.43	28.71	27.83	29.12	26.09	27.36	24.34	25.60
	32	26.33	30.19	28.16	33.59	28.55	33.17	26.48	30.64	25.21	29.78
		28.13	29.96	30.77	33.41	30.88	32.94	28.81	30.44	27.48	29.47
25	8	22.82	23.73	24.01	25.09	24.66	25.70	23.63	24.54	20.67	21.83
		23.62	24.56	25.27	26.30	25.85	26.83	24.47	25.42	22.05	23.04
	16	24.00	26.57	25.44	29.11	26.10	29.30	24.81	27.32	22.37	25.87
		25.55	26.84	27.80	29.23	28.15	29.48	26.35	27.63	24.68	25.99
	32	26.38	30.77	28.71	34.81	28.61	33.98	27.13	31.15	25.97	30.74
		28.72	30.45	31.67	34.57	31.63	33.67	29.37	30.87	28.28	30.54







(b) Online- ℓ_1 (25.69 dB)



(c) MLD-MulP (26.02 dB)



(d) RMLD-MulP (27.41 dB)

Fig. 9. Compressed recovery of images from random measurements (N = 8, SNR of measurement process = 15dB) using the different dictionaries. In each case the PSNR of the recovered image is also shown.

comparison, we use learned iterative dictionaries of size 1024, using ℓ_1 minimization in the SPAMS toolbox [19] and the Lagrangian dual method (SC-LD) [52]. Finally, classification was performed using a simple 1-nearest neighbor classifier. Table III and Table IV show the classification accuracies obtained using the different dictionaries, for both the subspace learning approaches. As it can be observed, graphs constructed with the proposed multilevel dictionaries provide more discriminative embeddings compared to the other approaches.

VI. CONCLUSIONS

We presented a multilevel learning algorithm to design generalizable and stable global dictionaries for sparse representations. The proposed algorithm uses multiple levels of 1–D subspace clustering to learn dictionaries. We also proposed a method to infer the number of atoms in each level, and provided an ensemble learning approach to create robust

dictionaries. We proved that the learning algorithm converges, exhibits energy hierarchy, and is also generalizable and stable. Finally, we demonstrated the superior performance of MLD in applications such as compressive sensing and subspace learning. Future research could include providing an online framework for MLD that can work with streaming data, and also developing hierarchical dictionaries that are optimized for robust penalties on reconstruction error.

REFERENCES

- D. J. Field, "What is the goal of sensory coding?" Neural Comp., vol. 6, pp. 559–601, 1994.
- [2] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, 2010
- [3] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," SIAM Review, vol. 43, no. 1, pp. 129–159, 2001.
- [4] M. Elad et.al., "A wide-angle view at iterated shrinkage algorithms," in SPIE, 2007.

TABLE III
UNSUPERVISED SUBSPACE LEARNING - CLASSIFICATION ACCURACIES
WITH A 1-NN CLASSIFIER.

# Train	Graph Construction Approach						
Per Class	LPP	SC-LD	SC-MLD	SC-RMLD			
250	57.11	57.5	58.2	58.9			
500	58.8	59.9	61.35	62.58			
1000	66.33	67.6	68.94	69.91			
1500	70.16	71.32	73.65	74.38			
2160	74.39	75.8	78.26	78.84			

TABLE IV SUPERVISED SUBSPACE LEARNING - CLASSIFICATION ACCURACIES WITH A 1-NN CLASSIFIER.

# Train	Graph Construction Approach						
Per Class	LDE	SC-LD	SC-MLD	SC-RMLD			
250	59.23	59.1	59.3	59.6			
500	60.4	60.8	61.9	62.7			
1000	68.1	68.71	69.6	70.43			
1500	72.9	73.5	74.41	75.09			
2160	77.3	78.07	79.53	80.01			

- [5] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE TSP*, vol. 54, no. 11, pp. 4311–4322, November 2006.
- [7] K. Huang and S. Aviyente, "Sparse representation for signal classification," in NIPS, 2006.
- [8] J. J. Thiagarajan, K. N. Ramamurthy, P. Knee, and A. Spanias, "Sparse representations for automatic target classification in SAR images," in ISCCSP, 2010.
- [9] J. Wright et.al., "Robust face recognition via sparse representation," IEEE TPAMI, vol. 31, no. 2, pp. 210–227, 2001.
- [10] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE CVPR*, Jun. 2010, pp. 3501 –3508.
- [11] J. Yang et.al., "Linear spatial pyramid matching using sparse coding for image classification," in IEEE CVPR, 2009.
- [12] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *IEEE ICIP*, Sep. 2010, pp. 1641 –1644.
- [13] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE CVPR*, 2010.
- [14] Z. Jiang et.al., "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *IEEE CVPR*, 2011.
- [15] J. J. Thiagarajan, K. N. Ramamurthy, P. Sattigeri, and A. Spanias, "Supervised local sparse coding of sub-image features for image retrieval," in *IEEE ICIP*, 2012.
- [16] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with ℓ_1 -graph for image analysis." *IEEE TIP*, vol. 19, no. 4, pp. 858–66, apr 2010.
- [17] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.
- [18] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comp.*, vol. 12, no. 2, pp. 337–365, 2000.
- [19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *JMLR*, vol. 11, no. 1, pp. 19–60, 2009.
- [20] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *ICML*, J. Frankranz and T. Joachims, Eds. Omnipress, 2010, pp. 487–494.
- [21] L. Bar and G. Sapiro, "Hierarchical dictionary learning for invariant classification," in *IEEE ICASSP*, March 2010, pp. 3578–3581.
- [22] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010
- [23] I. Tosic and P. Frossard, "Dictionary learning," IEEE Sig. Proc. Mag., vol. 28, no. 2, pp. 27–38, 2011.

- [24] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE PAMI*, vol. 34, no. 4, pp. 791–804, 2012.
- [25] Y. Zhou and K. Barner, "Locality constrained dictionary learning for nonlinear dimensionality reduction," *IEEE SP Letters*, 2012.
- [26] H. Wang, C. Yuan, W. Hu, and C. Sun, "Supervised class-specific dictionary learning for sparse modeling in action recognition," *Patt. Rec.*, vol. 45, no. 11, pp. 3902–3911, 2012.
- [27] I. Ramírez and G. Sapiro, "An MDL framework for sparse coding and dictionary learning," *IEEE TSP*, vol. 60, no. 6, pp. 2913–2927, 2012.
- [28] R. Gribonval and K. Schnass, "Dictionary Identification Sparse Matrix-Factorisation via ℓ₁-Minimisation," *IEEE Trans. on Inf. Theory*, vol. 56, no. 7, pp. 3523–3539, Jul. 2010.
- [29] Z. He et.al., "K-hyperline clustering learning for sparse component analysis," Sig. Proc., vol. 89, pp. 1011–1022, 2009.
- [30] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi, "General conditions for predictivity in learning theory," *Nature*, vol. 428, no. 6981, pp. 419– 422, 2004.
- [31] P. D. Grünwald, I. J. Myung, and M. A. Pitt, Advances in minimum description length: Theory and applications. MIT press, 2005.
- [32] S. Zhu, K. Shi, and Z. Si, "Learning explicit and implicit visual manifolds by information projection," *Patt. Rec. Letters*, vol. 31, pp. 667–685, 2010.
- [33] A. Gersho and R. Gray, Vector Quantization and Signal Compression. Boston: Kluwer Academic Publishers, 1992.
- [34] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *IEEE ICIP*, 2010.
- [35] A. Rakhlin and A. Caponnetto, "Stability of K-means clustering," in Advances in Neural Information Processing Systems, vol. 19. Cambridge, MA: MIT Press, 2007.
- [36] S. Ben-David, U. von Luxburg, and D. Pál, "A sober look at clustering stability," Conference on Computational Learning Theory, pp. 5–19, 2006.
- [37] S. Ben-David, D. Pál, and H. U. Simon, "Stability of K-means clustering," ser. Lecture Notes in Computer Science, vol. 4539. Springer, 2007, pp. 20–34.
- [38] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Optimality and stability of the K-hyperline clustering algorithm," *Patt. Rec. Letters*, 2010
- [39] A. Maurer and M. Pontil, "K-Dimensional Coding Schemes in Hilbert Spaces," *IEEE Trans. on Inf. Theory*, vol. 56, no. 11, pp. 5839–5846, 2010.
- [40] D. Vainsencher and A. M. Bruckstein, "The Sample Complexity of Dictionary Learning," JMLR, vol. 12, pp. 3259–3281, 2011.
- [41] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Multilevel dictionary learning for sparse representation of images," in *IEEE DSP Workshop*, 2011.
- [42] P. Agarwal and N. Mustafa, "K-means projective clustering," in ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2004, pp. 155–165.
- [43] P. Tseng, "Nearest q-flat to m points," Journal of Optimization Theory and Applications, vol. 105, no. 1, pp. 249–252, 2000.
- [44] A. Caponnetto and A. Rakhlin, "Stability properties of empirical risk minimization over Donsker classes," JMLR, vol. 7, pp. 2565–2583, 2006.
- [45] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion," Wavelets in Geophysics, vol. 4, pp. 299–324, 1994.
- [46] "Berkeley segmentation dataset," Available at http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/ segbench/.
- [47] L. Breiman, "Pasting small votes for classification in large databases and on-line," *Machine Learning*, vol. 36, no. 1, pp. 85–103, 1999.
- [48] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE PAMI*, vol. 29, no. 1, pp. 40–51, 2007.
- [49] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
- [50] X. He and P. Niyogi, "Locality preserving projections," in NIPS, 2003.
- [51] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *IEEE CVPR*, vol. 2, 2005, pp. 846–853.
- [52] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in NIPS, 2006, pp. 801–808.