A Novel Geometric Multiscale Approach to Structured Dictionary Learning on High Dimensional Data

Guangliang Chen
Department of Mathematics and Statistics
San José State University
One Washington Square
San José, California 95192-0103
Email: guangliang.chen@sjsu.edu

Abstract—Adaptive dictionary learning has become a hot-topic research field during the past decade. Though several algorithms have been proposed and achieved impressive results, they are all computationally intensive due to the lack of structure in their output dictionaries. In this paper we build upon our previous work and take a geometric approach to develop better, more efficient algorithms that can learn adaptive structured dictionaries. While inheriting many of the advantages in the previous construction, the new algorithm better utilizes the geometry of data and effectively removes translational invariances from the data, thus able to produce smaller, more robust dictionaries. We demonstrate the performance of the new algorithm on two data sets, and conclude the paper by a discussion of future work.

I. INTRODUCTION

Since the pioneering work of Olshausen and Field [1], adaptive dictionaries have quickly gained overwhelming popularity in the sparse representation and image processing communities, achieving state-of-the-art results in many imaging tasks, such as compression [2], denoising [3]–[5], deblurring [6], and super-resolution [7], to name a few.

Specifically, given training signals \(x_1, \ldots, x_n \in \mathbb{R}^\ell\), we learn from them a dictionary \(D = [d_1, \ldots, d_m] \in \mathbb{R}^{\ell \times m}\), which can be thought of as an overcomplete basis consisting of atomic signals \(d_i\) (called atoms). We then use the learned dictionary \(D\) to represent any signal \(x \in \mathbb{R}^\ell\) by a sparse coefficient vector \(\gamma\) that satisfies \(x = D\gamma\).

In practice, the dictionary learning task is often formulated as an optimization problem over both the dictionary \(D\) and corresponding coefficients \(\Gamma = [\gamma_1, \ldots, \gamma_n] \in \mathbb{R}^{m \times n}\):

\[
\min_{D, \Gamma} \sum_{i=1}^{n} \|x_i - D\gamma_i\|_2^2 + \lambda \|\gamma_i\|_0.
\]

(1)

Here, the term \(\|\gamma_i\|_0\) measures the sparsity of \(\gamma_i\) by counting the number of nonzeros in it, and \(\lambda\) is a regularization parameter balancing between representation accuracy and cost.

Starting this century, several algorithms [8]–[10] have been proposed for solving (1), most of which alternate between updating \(D\) and \(\Gamma\). That is, with some initial guess of the dictionary, they first fix it and consider the reduced problem over \(\Gamma\) and strive to find the best update for it (this part is referred to as sparse coding [11]–[13], a problem already extensively studied in the literature). Next, fixing the new coefficients \(\Gamma\), they then seek a new dictionary \(D\) that can decrease the objective function in (1). Such a two-step procedure is repeated until some stopping criterion has been reached.

Empirically, these algorithms have achieved very impressive results, often considerably outperforming transform-based methods (see e.g. [8]). However, a significant disadvantage of them is the lack of structure in their output dictionaries, making the sparse coding part extremely costly, especially when dealing with large data sets. Additionally, the success of such algorithms depends on the quality of the initial dictionary used, and often only convergence to a suboptimal solution is guaranteed. Consequently, there has been a pressing need for developing algorithms that can learn structured dictionaries from data [14].

In this paper, we propose a novel algorithm based on our previous work on Geometric Multi-Resolution Analysis (GMRA) [15]–[17]. The original GMRA framework already represents a step forward in this direction, by learning atoms at multiple spatial resolutions and organizing them into a tree. Nevertheless, it does not yield the most efficient dictionary because it neither uses “optimal partitions” of the data set, nor exploits any repeating pattern in the data geometry. To explain more specifically what these issues are, we first review in Section 2 the GMRA algorithm with some detail. Next, in Section 3, we describe our new strategies to resolve those problems and present a new algorithm. Section 4 concludes the paper by pointing out some future directions.

II. REVIEW OF THE GMRA FRAMEWORK

Assume a finite sample \(X = \{x_1, \ldots, x_n\}\) from a smooth compact Riemannian manifold \(M\) of dimension \(d\), isometrically embedded in \(\mathbb{R}^{\ell}\). We learn a hierarchical dictionary \(D\) using the following three steps:

(i). We perform a nested geometric decomposition of the data set \(X\) into dyadic cubes \(\{C_{j,k}\}_{0 \leq j \leq L, k \in \Gamma_j}\), with \(j\) representing scale (larger \(j\) corresponds to finer scale), \(k\) the
index of the cube $C_{j,k}$ at scale $j$, and $\Gamma_j$ the index set of all the cubes at scale $j$ (note that for each $j$ we have a disjoint partition $X = \bigcup_{k \in \Gamma_j} C_{j,k}$). This yields a tree whose nodes represent the dyadic cubes $C_{j,k}$ and each node is the union of its children. Practically, such a partition is obtained by using the METIS algorithm [18], and the finest scale $J$ is selected such that every leaf node can be approximated by a $d$-dimensional plane, or in short, $d$-plane, within a given level of precision $\epsilon$.

(ii). We apply principal component analysis (PCA) to every $C_{j,k}$ and form an orthonormal matrix $\Phi_{j,k} \in \mathbb{R}^{\ell \times d}$ using the top $d$ PCA vectors. We call $\Phi_{j,k}$ the scaling basis associated to $C_{j,k}$. The corresponding PCA subspace, denoted $S_{j,k} := \text{colspan}(\Phi_{j,k})$, represents a best $d$-plane approximating $C_{j,k}$ (after being centered) in the least square sense. We then project $C_{j,k}$ onto $S_{j,k}$: For a fixed $j$, the projections of the subsets $\{C_{j,k}\}_{k \in \Gamma_j}$ onto their respective PCA $d$-planes form a piecewise linear set $X_j$, representing the scale-$j$ approximation to $X$.

(iii). We construct, at every scale $j$, low-dimensional wavelet bases $\{\Psi_{j,k}\}_{k \in \Gamma_j}$ to encode the difference between $X_j$ and $X_{j-1}$. Specifically, for any $k \in \Gamma_j$ and the index $k'$ of its parent at scale $j-1$, let $\Psi_{j,k}$ be an orthonormal basis for the projection of $S_{j,k}$ onto the orthogonal complement of $S_{j-1,k'}$ in $\mathbb{R}^\ell$:

$$\text{colspan}(\Psi_{j,k}) := (I - \Phi_{j-1,k'}\Phi_{j-1,k'}^T) S_{j,k}$$

(2)

The collection $\{\Psi_{j,k}\}_{0 \leq j \leq J, k \in \Gamma_j}$ forms a hierarchically-organized dictionary that encodes the “details” of the data at various resolutions.

The GMRA construction naturally extends the wavelet transform for 1D signals and PCA for linear subspaces to efficient multiscale transforms for high-dimensional data sampled from a nonlinear manifold. It has many advantages: (i) The constructions of the scaling and wavelet bases use only rank-$d$ SVD and thus run fast. (ii) Theoretical guarantees may be derived on the size of the dictionary and the sparsity of the coefficients. Additionally, due to the intrinsic low-dimensional structure of the data, the required dictionary size is independent of the ambient dimension. (iii) The dictionary is associated with fast transforms that explicitly compute the sparse coefficients, which are typically unavailable in other methods. For more details about the construction, its advantages and variations (including how to deal with general point-cloud data), we refer the reader to [17].

Though theoretically very appealing, the original GMRA framework is not optimized for practical purposes for at least two reasons. First, GMRA divides a manifold into dyadic cubes, in a way similar to the Haar wavelets, to examine the differences between their projections. While this guarantees the wavelet coefficients to decay at a certain rate, such a partitioning scheme does not explicitly follow the geometry of the manifold. As a result, the multiscale partitions are often far from being optimal, leading to overly large dictionaries. Second, since GMRA always combines nearby dyadic cubes to form coarser decompositions of the data set, it is not efficient when the manifold contains very similar, yet faraway, segments (e.g., the top and bottom portions of an S-manifold), because it repeats the same construction and generates very similar dictionary atoms at those locations. It would be desirable to first combine them together for learning a common subset of atoms.

III. NEW ALGORITHM

In this section we describe our new strategies to address the above-mentioned issues.

A. New decomposition scheme

Given a data set with complex geometry, we propose to partition it into (disjoint) low-dimensional Gaussian clouds (at a given precision), which we believe are the atomic parts of the data, because they have no further sparse structure. We will then use these low-dimensional subsets as the leaf nodes of a new partition tree (their inner nodes will be defined in the next subsection). In general, such a partitioning task is computationally challenging for many reasons, such as large data size, high ambient dimension, heavy noise, unknown local dimensions and optimal partition size, and nonuniform sampling. Due to page limit and for simplicity, we consider only manifold data in this manuscript, leaving the general case to future work.

When the data is sampled from a $d$-dimensional manifold, the atomic parts are the local linear segments (patches) of the manifold. To focus on the new ideas, we assume that $d$ is known (otherwise, we may use the Multiscale SVD technique [19] to accurately infer it). Alg. 1 presents a randomized procedure to identify such fundamental structures in the data set.

Algorithm 1 Data partitioning via localized $K$-Planes

Input: Training data $X$, manifold dimension $d$, precision $\epsilon$

Output: A partition of $X$ into disjoint linear patches

Steps:

1: Randomly select a point $x \in X$ and find the largest neighborhood $N(x) \subset X$ around it, such that $N(x)$ can be approximated by a $d$-plane within precision $\epsilon$. Remove $N(x)$ from $X$ and repeat this procedure until $X$ becomes empty. Denote all the selected points by $\{c_g\}_{1 \leq g \leq G}$.

2: For each neighborhood $N(c_g)$, calculate its covariance matrix $\Sigma_g$. Then use them to compute the Mahalanobis distances between the training points $x_i \in X$ and neighborhood centers $c_g$:

$$\text{dist}^2(x_i, c_g) = (x_i - c_g)^T \Sigma_g^{-1} (x_i - c_g).$$

3: Reassign points in $X$ to their nearest neighborhood centers to form a new partition of $X$ (remove empty subsets, if any). Update the neighborhood centers $c_g$ by averaging.

4: Repeat Steps 2 and 3 until some convergence criterion has been reached (e.g., the total fitting error by $d$-planes stops decreasing).

5: Return the final neighborhoods $\{N(c_g)\}_{1 \leq g \leq G}$.
B. New merging scheme for building the partition tree

For the linear patches \( \{N(c_g)\}_{1 \leq g \leq G} \) obtained by Alg. 1, the original GMRA would recursively combine them based on their spatial proximity. We propose to merge them according to their “geometric similarities”. To define such a notion, we first subtract the centers \( c_g \) from \( N(c_g) \) to remove translations (we denote the centered patches by \( \tilde{N}(c_g) \)); another benefit is that this fully makes use of the centers in terms of their ability to approximate their associated patches. We will from now on focus on such centered data, or equivalently, residuals.

Let \( B_g \) represent an orthonormal basis for the PCA \( d \)-plane of \( \tilde{N}(c_g) \), for each \( g \). We note that the collection \( \{B_g\} \) can already be used as a dictionary, corresponding to a union of subspaces model [20], [21]. However, because atoms from different subspaces cannot jointly represent a signal, such a dictionary is still too restrictive. We aim to build a hierarchical dictionary from this collection of bases which is more efficient and can further sparsify data. For this goal, we compute a geometric similarity score between any pair \( \tilde{N}(c_{g_1}), \tilde{N}(c_{g_2}) \) using their PCA \( d \)-planes:

\[
A(g_1, g_2) = \exp\left(-\frac{1}{\sigma} \|B_{g_1}B_{g_1}^T - B_{g_2}B_{g_2}^T\|_F^2\right). \tag{4}
\]

Here, \( \| \cdot \|_F \) denotes the Frobenius norm and \( \sigma > 0 \) is a tuning parameter specified by the user. The collection of these scores forms a pairwise similarity matrix, which can be used by spectral clustering algorithms to group the centered patches. We choose the two-way spectral clustering algorithm of Shi and Malik [22] and recursively apply it to the collection \( \{\tilde{N}(c_g)\}_{1 \leq g \leq G} \) (with the above similarity measure), until all final clusters contain only one distinct \( \tilde{N}(c_g) \), to obtain a nested decomposition of the centered patches \( \{\tilde{N}(c_g)\} \); see Alg. 2.

**Algorithm 2** Patch merging using geometric similarities

**Input:** Patch bases \( \{B_g\}_{1 \leq g \leq G} \), tuning parameter \( \sigma \)

**Output:** A hierarchical clustering of \( \{\tilde{N}(c_g)\}_{1 \leq g \leq G} \)

**Steps:**
1. Form a \( G \times G \) similarity matrix \( A \) using (4).
2. Apply 2-way spectral clustering with \( A \) to partition the collection \( \{\tilde{N}(c_g)\} \) into two subcollections (clusters).
3. For each new cluster apply 2-way spectral clustering with the reduced \( A \) to further divide it. Repeat this procedure until every cluster contains exactly one patch.
4. **Return** the hierarchical clusters obtained above, encoded with a binary tree.

Comparing with the original GMRA, the new merging scheme has the following advantages. First of all, centering the linear patches effectively removes translational symmetries present in the data (at the patch scale). Secondly, (geometrically) similar data are moved together and will be used jointly for dictionary learning. As a result, this may considerably reduce the size of the dictionary, further sparsify the coefficients, and meanwhile increase the robustness of the algorithm. Lastly, the proposed merging scheme is global in nature. This is very similar to the Non-Local Means algorithm [3], which simultaneously considers all similar pixel neighborhoods within an image for more effective denoising. This simple idea has achieved very impressive results in practice because it fully utilizes the self-expressiveness of natural image content.

C. Multiscale adaptive dictionary learning

Incorporating the new partitioning and merging schemes presented earlier, we formulate a new algorithm for learning structured dictionaries from manifold data; see Alg. 3.

**Algorithm 3** Proposed new dictionary learning algorithm

**Input:** Training data \( X \), manifold dimension \( d \), precision \( \epsilon \)

**Output:** Wavelet bases \( \{\Psi_{j,k}\} \), patch centers \( \{c_g\} \) and bases \( \{B_g\} \)

**Steps:**
1. Apply Alg. 1 to partition \( X \) into linear patches \( \{N(c_g)\} \), and then compute their \( d \)-dimensional PCA bases \( \{B_g\} \).
2. Apply Alg. 2 to obtain a nested partition of the centered patches \( \{\tilde{N}(c_g)\} \), encoded with a binary tree. For each leaf node \((j, k)\), rename the associated patch basis \( B_g \) as \( \Phi_{j,k} \). Let \( J \) denote the largest depth of all leaf nodes.
3. Construct wavelet bases \( \Psi_{j,k} \) at every node of the tree by using a bottom-up procedure:
   - For each node \((j, k)\) at scale \( j \)
     - Compute the \( d \)-dimensional PCA basis \( \Phi_{j-1,k'} \) for the parent node \((j-1, k')\) using the corresponding union of centered patches.
     - Construct the wavelet basis \( \Psi_{j,k} \) using (2).

At the root node, set \( \Psi_{0,1} = \Phi_{0,1} \).
4. **Return** \( \{\Psi_{j,k}\}, \{c_g\}, \{B_g\} \).

The output dictionary consists of all wavelet bases \( \{\Psi_{j,k}\} \) and patch centers \( \{c_g\} \). For any test signal \( x \in \mathbb{R}^d \), we may use Alg. 4 to calculate the coefficients needed to represent it. The algorithm uses only one path in the tree – from the closest leaf node to the root of the tree – and works by recursively projecting \( x \) onto the subspaces spanned by the wavelet bases and removing the projections from \( x \). Thus, it runs extremely fast. Note that although the patch bases \( B_g \) are part of the input to Alg. 4, they should not be viewed as a portion of the dictionary. Rather, they are needed only by new data points to assign them to the nearest patches and project them onto the corresponding PCA subspaces.

Finally, if we are given the coefficients \( \{q_i\} \) (and the path \( P \)), we may easily reconstruct \( x \) by adding up all the wavelet components as well as the patch center \( c_g \):

\[
x = c_g + \sum_{(i,k) \in P} \Psi_{i,k} \cdot q_i \tag{5}
\]

D. Demonstrations

In Fig. 1 we use a toy data set, consisting of 1000 points sampled from a circle, to illustrate the steps of the proposed
Algorithm 4 Sparse coding for output dictionary from Alg. 3

**Input:** Dictionary $D = \{\Psi_{j,k}\} \cup \{c_g\}$, test point $x$, and patch bases $\{B_g\}$

**Output:** Coefficient vectors $\{q_i\}$ and path used $\mathcal{P}$

**Steps:**

1. Assign $x$ to the nearest patch, say $\tilde{N}(c_g)$, in terms of the Mahalanobis metric in (3), and then project it onto the corresponding PCA $d$-plane: $x \leftarrow B_g B_g^T (x - c_g)$
2. Let $(j, k) \leftarrow$ the leaf node containing $\tilde{N}(c_g)$.
   
   **Repeat**
   
   $q_j \leftarrow \Psi_{j,k}^T \cdot x$, $x \leftarrow x - \Psi_{j,k} \cdot q_j$, $(j, k) \leftarrow$ parent$(j, k)$
   
   **until** the root node has been visited.

3. **Return** the coefficient vectors $\{q_i\}_{i \geq 0}$ and the path $\mathcal{P}$ used, i.e., the sequence of nodes visited.

IV. FUTURE WORK

We described a new framework based on the existing GMRA [17] for learning multiscale adaptive dictionaries. While inheriting many of the previous advantages (e.g., multiscale dictionary and fast transform), the new construction follows the manifold geometry more closely, makes use of the patch centers more efficiently, and can remove translational symmetries present in the data. As a result, the output dictionary is more efficient and reliable.

For simplicity, we focused on the manifold case in this manuscript and presented only the most basic ideas used in the new construction. In future research we plan to implement more sophisticated techniques and extend the current algorithms to handle more complex data. We mention below some new directions that we want to explore:

(i) We will extend Algs. 1-3 to deal with non-manifold data, in which case we believe that the atomic parts are the local Gaussian distributions that may be of different and unknown dimensions and densities. To partition such data, we need to incorporate the Multiscale SVD technique [19] into Alg. 1 to accurately and reliably detect local dimension and scale at random locations. We also believe that this mixture of Gaussians model has a close tie to the Bayesian approach [9], and we plan to investigate it.

(ii) After making the above extensions, we will further test our algorithms on the MNIST handwritten digits [23] and eventually apply them to challenging imaging tasks such as denoising and inpainting, hoping to compare them with the currently best empirical algorithms (e.g. [8]).

(iii) The proposed research actually corresponds to dictionary learning on the Grassmannian $\text{Gr}(d, \ell)$: If we regard the patch bases $\{B_i\}$ as points on $\text{Gr}(d, \ell)$, then both the original GMRA [17] and the proposed new algorithm use some sort of recursive “averaging and subtracting” of points on $\text{Gr}(d, \ell)$ to learn a multiscale dictionary. However, the “metrics” used by them are different: the original GMRA uses the spatial closeness of associated patches while the new algorithm uses their geometric similarity (which is a true metric on $\text{Gr}(d, \ell)$). We are currently working on applying GMRA in general non-Euclidean spaces and expect the new findings to guide us in developing more efficient and accurate dictionaries.

REFERENCES


Two remarks: (1) the patch centers can be regarded as the scale-0 approximation of the circle; (2) the leaf nodes of the tree correspond to the individual centered patches, and parallel ones are first merged at next coarser level. Middle row shows the approximations of the circle at different levels of the tree (from scale 1 to 4) obtained by the proposed new algorithm. Note that there is little difference between the reconstructions at levels 3 and 4, which is because all level-3 nodes consist of nearly parallel patches. The wavelet coefficients at scale 4 are all very small (thus compressible), and hence only the top three levels are needed for achieving the same precision. Accordingly, the required dictionary size (for achieving the same precision) is reduced by half, comparing with the original GMRA. Bottom row: the first three plots show the approximations of the circle by the original GMRA at scales 1 to 3 of another binary tree built with its own merging scheme (but using the same patches as leaf nodes). Note that at scale 4 it has exactly the same reconstruction of the circle as the new algorithm (thus not shown here), as both correspond to PCA approximations of the leaf node patches. However, the new algorithm uses both the patch centers and the dictionary atoms much more efficiently. As a result, the coarse-scale approximations of the proposed new algorithm are much more accurate; this is also obvious by comparing their mean $L_2$ approximation errors at all scales (last plot).