Geometric Manifold Approximation using Union of Tangent Patches

Talal Ahmed and Waheed U. Bajwa
{talal.ahmed, waheed.bajwa}@rutgers.edu
Dept. of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854

***Abstract*— This paper addresses the problem of data-adaptive learning of the ambient geometry of a nonlinear, non-intersecting submanifold of a Euclidean space. It accomplishes this goal by exploiting the local linearity of the (sub)manifold and approximating it using a *union of tangent patches* (UoTP). In addition, it translates the problem of projecting a new data point onto the learned UoTP into a series of convex optimization problems. It then derives a procedure for encoding (projecting) data points onto a UoTP that involves an efficient solution to each of the posed optimization problems. Finally, it demonstrates the value of capturing the geometry of manifolds by comparing the superior denoising performance of the proposed framework on both synthetic and real data sampled from nonlinear manifolds with that of stat-of-the-art denoising algorithms.**

***Index Terms*—Data encoding, denoising, manifold learning, tangent approximations**

I. Introduction

Data models play an increasingly important role in information processing. Owing to computational advances of recent years, classical linear models are slowly but surely being replaced by their nonlinear generalizations. The union-of-subspaces (UoS) model and the (nonlinear) manifold model in particular stand out among popular nonlinear data models [1]. In order to incorporate any nonlinear model into most information processing tasks, one requires scalable algorithms for (i) data-adaptive learning of geometric structure of the underlying model (structure learning) and (ii) projecting arbitrary data onto the learned geometric structure (data encoding). Significant progress has been made in this regard for the UoS model under the rubrics of *dictionary learning* and *sparse coding* [2]–[4]. But relatively less progress has been made toward these two goals for the manifold model. Our focus in this paper is on the (nonlinear) manifold model and we present novel algorithms for structure learning and data encoding (i.e., projection/representation) in the following in this nonlinear setting.

Relationship to prior work: While manifold data models have been considered in the literature for more than a decade now, most of that work focuses on dimensionality reduction and learns only low-dimensional embedding of data [5]–[7]. Many other works that focus on learning manifold structure in the ambient space assume parametric manifolds [8]. Some recent works have tried to remedy these shortcomings [9]–[12]. But the structures learned in [10], [11] do not fully adapt to the underlying manifold curvature, while [10], [12] lack algorithms for data encoding.

Finally, note that one can in principle use nonlinear models such as the UoS model (and its variants such as the *hybrid linear model* [13], [14]) for learning the structure of data sampled from a nonlinear manifold. However, while such an approach will lead to small approximation error for the training data, it will fail to capture the exact geometry of the underlying manifold (cf. [12, Fig. 1(b)]). One of the goals of this paper in this regard is demonstrating that capturing the geometry of data is as important as approximating the data.

Our contributions: In this paper, we present algorithms for (i) learning the structure of data sampled from a nonlinear, nonparametric submanifold of a Euclidean space, and (ii) encoding (projection) of given data onto the learned structure. Instead of focusing on topological aspects and/or dimensionality reduction, we focus on learning the manifold structure in the ambient space—an approach that we term *geometric manifold approximation* (GMA) to differentiate it from traditional *manifold learning* approaches [5]–[7]. Our approach to GMA involves exploiting the local linearity of a manifold and approximating it using a *union of tangent patches* (UoTP). Note that [9]–[12] also make use of local linearity of manifolds and approximate them using unions of *tangent flats*. However, the tangent flats in [9]–[12] extend indefinitely in the ambient space, which complicates the data encoding procedure. In contrast, our use of tangent patches allows us to transform the problem of projecting data onto a UoTP into a series of convex programs and we present efficient means of solving these convex programs. Finally, we underscore the importance of our proposed algorithms by demonstrating their denoising performances on both synthetic and real data.

II. Problem Formulation

We are interested in the problem of approximating the geometry of a smooth d-dimensional Riemannian manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$ using a collection of tangent patches. In addition, we need means of projecting a new data sample $x \in \mathbb{R}^D$ onto the learnt collection of tangent patches. In order to learn $\mathcal{M}$, we are given a collection of N training data points sampled from $\mathcal{M}$: $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathcal{M}$. Our main assumptions here are that $\mathcal{M}$ is not self-intersecting, and the manifold dimension $d \ll D$ is known a priori; see [15] for possible means of estimating d from the data.

Mathematically, our goal is to use $\mathcal{X}$ to learn a set of tangent patches, $\mathcal{T}$, that provides a reasonable approximation of the data in $\mathcal{X}$. Specifically, let $\mathcal{A}$ be a set that indexes the set of tangent patches, $\mathcal{T}$. Then $\mathcal{T}$ can be mathematically expressed as $\mathcal{T} = \{\mathcal{T}_k\}_{k\in\mathcal{A}}$, where each tangent patch $\mathcal{T}_k$ is defined as $\mathcal{T}_k = \{\phi_k, c_k, r_k, r'_k, \mathcal{C}_k\}$. Here, $\phi_k \in \mathbb{R}^{D\times d}$ denotes an orthonormal basis matrix that describes a centered tangent plane, $c_k \in \mathbb{R}^D$ denotes offset of this tangent plane from the origin, $r_k, r'_k \in \mathbb{R}^D$ denote parameters that help define the subset of the tangent plane that forms the $k \in \mathcal{A}$ tangent patch, and $\mathcal{C}_k \subset \mathcal{X}$ is the training data associated with this k-th patch. Concretely, the set of points in tangent patch $\mathcal{T}_k$, $k \in \mathcal{A}$, can be expressed using $\{\phi_k, c_k, r_k, r'_k\}$ as:

$$\widehat{\mathcal{M}}_k = \{y : y = \phi_k w + c_k, r_k \preccurlyeq y \preccurlyeq r'_k, w \in \mathbb{R}^d\},\ k \in \mathcal{A}, \quad (1)$$

where $\preccurlyeq$ represents component-wise inequality. We can use this notation to express the union of tangent patches (UoTP) approximation of $\mathcal{M}$ as $\widehat{\mathcal{M}} = \bigcup_{k\in\mathcal{A}} \widehat{\mathcal{M}}_k$.

We are now ready to formalize the problem of geometric manifold approximation (GMA). Given $\mathcal{X} \subset \mathcal{M}$, we need to find a UoTP-based GMA $\widehat{\mathcal{M}}$ of the manifold $\mathcal{M}$ such that the approximation

error e_k associated with each tangent patch $\mathcal{T}_k$, $k \in \mathcal{A}$, is within ϵ:

$$e_k = \frac{1}{|\mathcal{C}_k|} \sum_{x \in \mathcal{C}_k} \frac{\|x - \phi_k \phi_k^\top (x - c_k) - c_k\|}{\|x - c_k\|} \leq \epsilon, \ k \in \mathcal{A}, \quad (2)$$

where $\|.\|$ denotes the Euclidean distance. In parallel, our goal is to minimize the number of tangent patches, $|\mathcal{A}|$, needed to approximate $\mathcal{M}$ up to the accuracy described by (2). The rationale for this objective is that it will reduce storage costs, data encoding complexity, etc. Stated differently, our goal in this paper is to minimize the number of tangent patches needed to approximate the geometry of $\mathcal{M}$ up to a certain accuracy level. Such an objective effectively makes the final number of patches in an approximation adaptive to the manifold curvature, which also addresses the problem of finding an appropriate number of linear structures required for manifold approximation in [9]–[11]. In the following, we present our approach to achieving this objective. This approach will be termed *geometry preserving union-of-tangent-patches* (GP-UoTP) learning to highlight the fact that the learned collection of tangent patches is not an arbitrary one; rather, it accurately captures the local manifold geometry.

III. Approximating Manifold Geometry using a Union of Tangent Patches

In this section, we first derive our algorithm for GP-UoTP learning of the manifold $\mathcal{M}$ (Sec. III-A). This will be followed by our formulation of the data encoding/projection problem and our derivation of an efficient solution to that problem (Sec. III-B).

A. GP-UoTP-based learning of (nonlinear) manifolds

To begin, we associate with each data sample $x \in \mathcal{X}$ a neighborhood set $\mathcal{N}_K(x)$, defined as a set of K nearest neighbors of x in $\mathcal{X}$ with respect to the Euclidean distance. Next, we initialize our algorithm by associating with each $x \in \mathcal{X}$ a tangent patch $\mathcal{T}_x$. This initial tangent patch $\mathcal{T}_x$ is characterized by an orthonormal basis matrix $\phi_x \in \mathbb{R}^{D \times d}$, obtained via *singular value decomposition* (SVD) of all the points in $\mathcal{N}_K(x)$, a vector $c_x \in \mathbb{R}^D$, which is the mean of all the samples in $\mathcal{N}_K(x)$, and a training subset $\mathcal{C}_x$, which is initialized as $\mathcal{C}_x = \{x\}$ to ensure all tangent patches have disjoint training sets. The choice of parameters $r_x, r'_x \in \mathbb{R}^D$ in this initial phase is immaterial and we arbitrarily set them to $r_x = r'_x = x$. After initialization, therefore, we obtain a collection of tangent patches $\mathcal{T}$ given by: $\mathcal{T} = \{\mathcal{T}_j\}_{j \in \mathcal{A}}$, where $\mathcal{T}_j = \{\phi_j, c_j, r_j, r'_j, \mathcal{C}_j\}$ and the set $\mathcal{A}$ indexes the set of patches in $\mathcal{T}$ with $|\mathcal{A}| = |\mathcal{X}|$ at this stage.

Our goal now is to minimize the number of tangent patches in $\mathcal{T}$ while satisfying the constraint in (2). We adopt a greedy procedure for this and greedily merge/fuse pairs of tangent patches in $\mathcal{T}$ till any further fusion of any pair of tangent patches in $\mathcal{T}$ violates (2). During this procedure, however, we must ensure that only neighboring tangent patches are fused together. In this regard, we leverage the notion of *fusibility of clusters* in [10] and define the set of all pairs of fusible patches in $\mathcal{T}$ as

$$\Omega = \{(i,j) \in \mathcal{A} \times \mathcal{A} : y_i \in \mathcal{C}_i, y_j \in \mathcal{C}_j \text{ such that } y_i \in \mathcal{N}_K(y_j) \text{ or } y_j \in \mathcal{N}_K(y_i)\}. \quad (3)$$

In order to greedily fuse pairs of patches, we define the best fusible pair as the one which gives the least approximation error smaller than ϵ for the associated training dataset after fusion, i.e.,

$$(i^*, j^*) = \underset{(i,j) \in \Omega}{\arg\min}\ e_{proj}(\mathcal{T}_i, \mathcal{T}_j) \text{ such that } e_{proj}(\mathcal{T}_i, \mathcal{T}_j) \leq \epsilon, \quad (4)$$

where the "fused" approximation error $e_{proj}(\mathcal{T}_i, \mathcal{T}_j)$ is defined as

$$e_{proj}(\mathcal{T}_i, \mathcal{T}_j) = \frac{1}{|\mathcal{C}_k|} \sum_{x \in \mathcal{C}_k} \frac{\|x - \phi_k \phi_k^\top (x - c_k) - c_k\|_2}{\|x - c_k\|_2}. \quad (5)$$

To understand the notation in (5), let us use $\mathcal{T}_k$ to denote the tangent patch obtained by merging $\mathcal{T}_i$ and $\mathcal{T}_j$. Then, $\mathcal{C}_k = \mathcal{C}_i \cup \mathcal{C}_j$ is the set of training data associated with $\mathcal{T}_k$, c_k is the empirical mean of all the samples in $\mathcal{C}_k$, and ϕ_k corresponds to the d dominant eigenvectors of the matrix $U_k = \frac{1}{2}(\phi_i \phi_i^\top + \phi_j \phi_j^\top)$.[1] Note that the matrix U_k can also be defined as in terms of an optimization problem:

$$U_k = \underset{U \in \mathbb{R}^{D \times D}}{\arg\min}\ \frac{1}{2}(\|\phi_i \phi_i^\top - U\|_F^2 + \|\phi_j \phi_j^\top - U\|_F^2). \quad (6)$$

Finally, we merge the best pair of patches $(\mathcal{T}_{i^*}, \mathcal{T}_{j^*})$ to obtain a new tangent patch $\mathcal{T}_{k^*}$, resulting in a reduced-cardinality set of patches:

$$\mathcal{T} = (\mathcal{T} \setminus \{\mathcal{T}_{i^*}, \mathcal{T}_{j^*}\}) \cup \{\mathcal{T}_{k^*}\}, \quad \text{and}$$
$$\mathcal{A} = (\mathcal{A} \setminus \{i^*, j^*\}) \cup \{k^*\}. \quad (7)$$

Once we carry out this greedy merging of one pair of patches and redefine our collection of tangent patches $\mathcal{T}$ and the corresponding index set $\mathcal{A}$, we again return to evaluation of (3), followed by the optimization problem (4) and the redefinition of $\mathcal{T}$ and $\mathcal{A}$ in (7). We repeat this procedure till the feasible set in (4) becomes empty. Finally, once that happens, we re-evaluate the parameters r_k, r'_k to completely characterize the subset of the tangent plane $\{\phi_k w + c_k\}, w \in \mathbb{R}^d, k \in \mathcal{A}$, that describes the tangent patch $\mathcal{T}_k$. Specifically, we define these parameters as $r_{k,i} = \min_{x \in \mathcal{C}_k} x_i$, and $r'_{k,i} = \max_{x \in \mathcal{C}_k} x_i$, $i \in \{1, ..., D\}$, $k \in \mathcal{A}$. Here, $r_{k,i}, r'_{k,i}$, and x_i denote the i-th element of r_k, r'_k, and x, respectively. After updating our collection of tangent patches with these parameters, we can approximate the manifold $\mathcal{M}$ in terms of the following GP-UoTP:

$$\widehat{\mathcal{M}} = \bigcup_{k \in \mathcal{A}} \{y : y = \phi_k w + c_k, r_k \preccurlyeq y \preccurlyeq r_k^{'}, w \in \mathbb{R}^d\}. \quad (8)$$

Our complete GP-UoTP learning procedure is outlined in Algorithm 1. We conclude by pointing out that the level at which $\widehat{\mathcal{M}}$ approximates $\mathcal{M}$ depends on the parameter ϵ: the smaller the value of ϵ, the more accurate the approximation and vice versa.

B. Projecting new data onto a UoTP

Our next goal is to develop a computational approach for projecting a new (possibly noisy) data sample $x \notin \mathcal{X}$ onto the learned structure $\widehat{\mathcal{M}}$. Mathematically, we are interested in solving the following problem: $\widehat{x} = \arg\min_{z \in \widehat{\mathcal{M}}} \|x - z\|, x \in \mathbb{R}^D$. This can be accomplished in principle by first projecting the sample x onto each of the tangent patches $\widehat{\mathcal{M}}_k$, $k \in \mathcal{A}$, identifying the patch $\widehat{\mathcal{M}}_{k^*}, k^* \in \mathcal{A}$, that results in the smallest projection error, and then defining $\widehat{x} = \arg\min_{z \in \widehat{\mathcal{M}}_{k^*}} \|x - z\|$. The first two steps of this procedure correspond to the following two problems:

$$\forall k \in \mathcal{A}, \ w_k = \underset{w \in \mathbb{R}^d}{\arg\min} \|\phi_k w + c_k - x\|_2^2 \quad \text{subject to}$$
$$r_k \preccurlyeq \phi_k w + c_k \preccurlyeq r'_k, \ \text{and} \quad (9)$$
$$k^* = \underset{k \in \mathcal{A}}{\arg\min} \|\phi_k w_k + c_k - x\|_2^2. \quad (10)$$

Once we have identified the index k^* using (10), we can simply write the final projection step as $\widehat{x} = \phi_{k^*} w_{k^*} + c_{k^*}$. In term of encoding

[1] An alternative strategy for defining U_k is to take a weighted mean of the projection matrices $\phi_i \phi_i^\top$ and $\phi_j \phi_j^\top$.

Algorithm 1: Learning Geometry-Preserving Union of Tangent Patches (GP-UoTP)

Input: Dataset: $\mathcal{X} \subset \mathcal{M} \subset \mathbb{R}^D$; manifold dimension: $d \ll D$; neighborhood size: K; maximum approximation error: ϵ

Initialization of collection of tangent patches

$\forall x \in \mathcal{X},\ \mathcal{N}_K(x) \leftarrow \{K \text{ nearest neighbors of } x \text{ in } \mathcal{X}\}$

$\forall x \in \mathcal{X},\ c_x \leftarrow \frac{1}{|\mathcal{N}_K(x)|} \sum_{y \in \mathcal{N}_K(x)} y$

$\forall x \in \mathcal{X},\ [\mathcal{N}_x^0] \in \mathbb{R}^{D \times |\mathcal{N}_K(x)|} \leftarrow \{y - c_x : y \in \mathcal{N}_K(x)\}$

$\forall x \in \mathcal{X},\ \phi_x \in \mathbb{R}^{D \times d} \leftarrow$ left singular vectors of $[\mathcal{N}_x^0]$ corresponding to its d-largest singular values

$\forall x \in \mathcal{X},\ r_x \leftarrow x, r'_x \leftarrow x,$ and $\mathcal{C}_x \leftarrow \{x\}$

Let $\mathcal{A} : |\mathcal{A}| = |\mathcal{X}|$ be a set that indexes all tangent patches in the collection $\mathcal{T} \leftarrow \{\mathcal{T}_j\}_{j \in \mathcal{A}}$ with $\mathcal{T}_j \leftarrow \{\phi_j, c_j, r_j, r'_j, \mathcal{C}_j\}$

Merging of the tangent patches

loop

  $\Omega \leftarrow \{(i,j) \in \mathcal{A} \times \mathcal{A} : y_i \in \mathcal{C}_i, y_j \in \mathcal{C}_j$ such that $y_i \in \mathcal{N}_K(y_j)$ or $y_j \in \mathcal{N}_K(y_i)\}$

  $(i^*, j^*) \leftarrow \arg\min_{(i,j) \in \Omega} e_{proj}(\mathcal{C}_i, \mathcal{C}_j)$

  if $e_{proj}(\mathcal{C}_{i^*}, \mathcal{C}_{j^*}) < \epsilon$ **then**

    $\mathcal{T}_{k^*} \leftarrow \mathsf{merge}(\mathcal{T}_{i^*}, \mathcal{T}_{j^*}), \mathcal{A} \leftarrow (\mathcal{A} \setminus \{i^*, j^*\}) \cup \{k^*\},$ and $\mathcal{T} \leftarrow (\mathcal{T} \setminus \{\mathcal{T}_{i^*}, \mathcal{T}_{j^*}\}) \cup \{\mathcal{T}_{k^*}\}$

  else

    break the **loop**

  end if

end loop

$\forall j \in \mathcal{A},\ r_j = \{\min_{x \in \mathcal{C}_j} x_i\}_{i=1}^D$, and $r'_j = \{\max_{x \in \mathcal{C}_j} x_i\}_{i=1}^D$

Output: A final set of patches: $\mathcal{T}_j = \{\phi_j, c_j, r_j, r'_j, \mathcal{C}_j\}_{j \in \mathcal{A}}$

requirements, note that we need to store a total of $d+1$ parameters, given by (k^*, w_{k^*}), to represent any $x \in \mathbb{R}^D$ in terms of a GP-UoTP.

It can be seen from this discussion that our problem of projection onto a GP-UoTP is effectively reduced to solving a series of convex optimization problems, one for each $k \in \mathcal{A}$, described by (9). In order to solve these problems, notice that each tangent patch $\widehat{\mathcal{M}}_k, k \in \mathcal{A}$, can be written as the intersection of an affine subspace B_k, defined as $B_k = \{y : y = \phi_k w + c_k, w \in \mathbb{R}^d\}$, and a polyhedron D_k, defined as $D_k = \{y : r_k \preccurlyeq y \preccurlyeq r'_k\}$. Accordingly, (9) can be replaced with the problem of projection of x onto the intersection of convex sets B_k and D_k. Note that while this hints toward the possible use of projection onto convex sets methods [16], such methods only guarantee a point—instead of the closest point to a given sample—in the intersection of closed convex sets.

In order to motivate our solution to the problem of projection of x onto $B_k \cap D_k$, we define

$$f_k(x) = \begin{cases} 0, & x \in B_k, \\ +\infty, & \text{otherwise,} \end{cases} \quad \text{and} \tag{11}$$

$$g_k(x) = \begin{cases} 0, & x \in D_k; \\ +\infty, & \text{otherwise.} \end{cases} \tag{12}$$

Using this notation, we can replace (9) and (10) with the following two problems:

$$\forall k \in \mathcal{A},\ y_k = \arg\min_{y \in \mathbb{R}^D} f_k(y) + g_k(y) + \|y - x\|_2^2, \text{ and} \tag{13}$$

$$k^* = \arg\min_{k \in \mathcal{A}} \|y_k - x\|_2^2. \tag{14}$$

Note that since $f_k(y) + g_k(y) \neq \infty$ only when $y \in B_k \cap D_k$, it is easy to convince oneself that (13) results in projection of x onto the set $\widehat{\mathcal{M}}_k = B_k \cap D_k$. We also have from [17] that (13) has a unique solution, given by $y_k = prox_{f_k + g_k}(x)$ with $prox_{f_k+g_k}(\cdot)$ denoting the proximity operator of the sum of the functions $f_k(\cdot)$ and $g_k(\cdot)$. Note that $prox_{f_k+g_k}(\cdot)$ can be considered a generalization of the notion of projection onto the underlying convex set $B_k \cap D_k$.

We now need means of computing $prox_{f_k+g_k}(\cdot)$. To the best of our knowledge, $prox_{f_k+g_k}(\cdot)$ cannot be calculated directly for our choice of $f_k(\cdot)$ and $g_k(\cdot)$. Nonetheless, we can use one of the proximal methods described in [17], which is termed Dykstra's projection algorithm [18], to split the problem of projecting onto $B_k \cap D_k$ into a series of projections onto the convex sets B_k and D_k separately. This is useful since the projection operators for B_k and D_k have closed-form expressions. Specifically, the projection operator $P_{B_k}(\cdot)$ for set B_k can be expressed as:

$$P_{B_k}(z) = \phi_k \phi_k^\top (z - c_k) + c_k, \quad z \in \mathbb{R}^D. \tag{15}$$

Similarly, the projection operator $P_{D_k}(\cdot)$ for set D_k is defined as:

$$P_{D_k}(z) = \{v_i\}_{i=1}^D, \quad z \in \mathbb{R}^D, \tag{16}$$

where

$$v_i = \begin{cases} z_i, & r_{k,i} \leq z_i \leq r'_{k,i}; \\ \arg\min_{q \in \{r_{k,i}, r'_{k,i}\}} |z_i - q|, & \text{otherwise.} \end{cases}$$

Using this notation, we now outline Dykstra's projection algorithm for projecting $x \in \mathbb{R}^D$ onto $\widehat{\mathcal{M}}_k = B_k \cap D_k$ in Algorithm 2.

Algorithm 2: Dykstra's Projection Algorithm

Initialize: $x_0 = x$, $p_0 = 0$, $q_0 = 0$

for $n = 0, 1, 2, \ldots$ **do**

  $y_n = P_{B_k}(x_n + p_n)$

  $p_{n+1} = x_n + p_n - y_n$

  $x_{n+1} = P_{D_k}(y_n + q_n)$

  $q_{n+1} = y_n + q_n - x_{n+1}$

end for

Note that the sequence $\{x_n\}, n = 1, 2, 3, \ldots,$ generated by Dykstra's algorithm is guaranteed to converge to the projection of x onto $\widehat{\mathcal{M}}_k$ [19]. We have now described means of efficiently solving (13) and (14), which return the index of the "encoding patch" k^* and the projection of $x \in \mathbb{R}^D$ onto the encoding patch, given by $y_{k^*} \in \mathbb{R}^D$. Finally, the encoding coefficients $w_{k^*} \in \mathbb{R}^d$ in this setting can be calculated from these parameters as $w_{k^*} = \phi_{k^*}^\top (y_{k^*} - c_{k^*})$.

IV. Numerical Results

In order to demonstrate value of the manifold learning and data encoding framework developed in this paper, we perform denoising experiments for manifold-sampled noisy data. In these experiments, we focus on both synthetic data, which helps us appreciate the value of geometry preservation in manifold learning, and real data. In the case of synthetic data, the (noiseless) training data $\mathcal{X}$ is generated by sampling 1200 data points uniformly at random from 3 half-turns of a Swiss roll in $\mathbb{R}^3$. We then use Algorithm 1 to learn a GP-UoTP approximation of $\mathcal{X}$. For comparison purposes, we use the K-SVD algorithm [4] to approximate $\mathcal{X}$ using an overcomplete dictionary with 20 atoms and sparse codes with with at most 2 nonzero entries.

Next, we sample another $N = 1200$ *test* data points uniformly at random from 3 half-turns of the same Swiss roll. We collect these

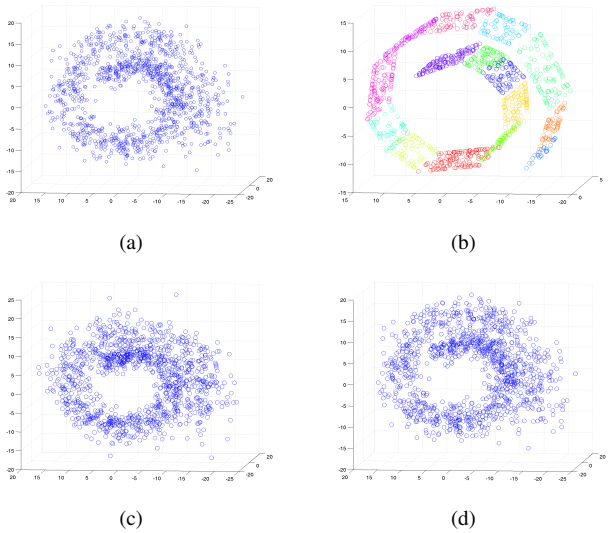

Fig. 1. (a) Noisy data sampled from 3-half turns of a Swiss roll (SNR = 10 dB). Denoised data using (b) GP-UoTP (MSE = 9.69 dB), (c) K-SVD (MSE = 11.88 dB) and (d) Haar wavelet thresholding (MSE = 10.98 dB).

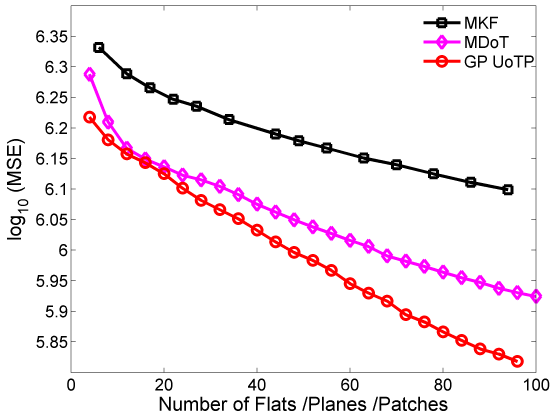

Fig. 2. Noise is added to 1000 images of digit 0 extracted from the MNIST database (SNR = 10dB). Noisy data is projected onto the learnt flats/planes/patches for denoising.

test samples into a matrix $Y \in \mathbb{R}^{3\times N}$ and then add additive white Gaussian noise to Y to obtain noisy test data Y_n (see Fig. 1(a)). In these experiments, we set the noise variance to be such that the *signal-to-noise ratio* (SNR) is 10 dB, where SNR $= 10\log\frac{\|Y\|_F^2}{\mathbb{E}[\|Y-Y_n\|_F^2]}$. We then denoise Y_n using our GP-UoTP framework as follows: we project each sample in Y_n onto our GP-UoTP approximation using the procedure of Sec. III-B to obtain denoised data $\widehat{Y}$. In addition, we also denoise Y_n using the dictionary obtained through K-SVD by sparsely coding the samples in Y_n via *orthogonal matching pursuit* [20]. Finally, we also denoise Y_n using wavelet denoising, which involves working with the Haar wavelet at a scale of 2 and a hard threshold of 1.26. Note that we chose these values for the scale and threshold after an exhaustive search over the choice of parameters for the best possible denoising results.

Our denoising results for GP-UoTP, K-SVD, and wavelet denoising are reported in Figs. 1(b), 1(c), and 1(d), respectively. It can be seen from these figures that denoising using our framework preserves the geometric structure of the original data, which can be attributed to our focus on geometry preservation. We also report quantitative results

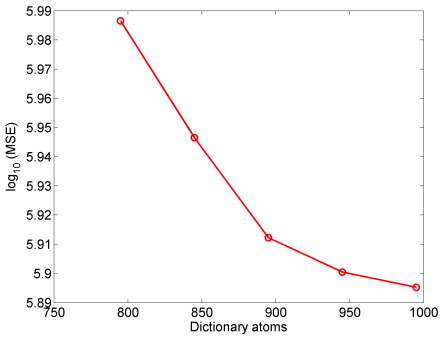

Fig. 3. Noise is added to 1000 images of digit 0 extracted from the MNIST database (SNR = 10dB). For denoising, the noisy data is sparse coded on the dictionary learnt using K-SVD.

for denoising using the metric of mean squared error (MSE), defined as MSE $= \frac{1}{N}\|Y-\widehat{Y}\|_F^2$. While the MSE of our denoised data is 9.69 dB, it is 11.88 dB for K-SVD and 10.98 dB for wavelet denoising.

In the case of denoising using real-world data, we focus on the MNIST database [21]. The noiseless data in this case corresponds to random sampling of $N = 1000$ images of digit 0 from the database. We vectorize each one of these images and then column-wise arrange them into a matrix Y. We then use GP-UoTP (our method), *merging based on difference of tangents* (MDoT) [9], and the *median K-flats* (MKF) [22] algorithms to approximate the geometry underlying Y in terms of a UoTP, a union of tangent planes, and a union of affine spaces, respectively. In all these algorithms, we select $d = 5$ for the dimension of the tangent patches/tangent planes/affine spaces. Similar to the case of synthetic data, we add noise to Y to obtain noisy data Y_n with SNR $= 10$ dB. We then denoise this noisy Y_n by projecting each sample in it onto its corresponding patch/plane/flat to obtain $\widehat{Y}$.[2] We repeat this learning and denoising procedure for varying number of patches, planes, and flats, respectively, so that the MSE results for these algorithms can be plotted as functions of number of patches/planes/flats. Our final results, averaged over 50 noise realizations, are reported in Fig. 2, which illustrate the superiority of our framework over the other two approaches.

We conclude this discussion by also reporting the denoising performance of K-SVD on digit '0' images. In this case, we set the sparsity to 5 and then report the K-SVD denoising performance in Fig. 3 as we vary the number of dictionary atoms. (Note that the number of dictionary atoms has to be less than $N = 1000$ for meaningful results.) Comparing Fig. 2 with Fig. 3 shows that GP-UoTP based denoising with large enough number of patches has better performance than K-SVD based denoising.

V. Conclusion

In this paper, we revisited the problem of data-adaptive learning of the ambient geometry of a nonlinear, non-intersecting submanifold of a Euclidean space. In this regard, we developed a framework that approximates the geometry of a nonlinear manifold from training data using a union of tangent patches (UoTP). In addition, we developed an efficient approach to projecting/encoding new data points onto the UoTP learned from training data. Finally, we demonstrated the value of our learning and encoding frameworks by comparing their denoising performance on both synthetic and real data with that of state-of-the-art methods in the literature.

[2]Note that we rely on our framework of Sec. III-B for these projections, whereas we have to rely on side knowledge for both MDoT and MKF due to the lack of explicit projection algorithms developed in [9] and [22].

REFERENCES

[1] R. G. Baraniuk, V. Cevher, and M. B. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proc. of the IEEE*, vol. 98, no. 6, pp. 959–971, 2010.

[2] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003.

[3] J. Mairal et al., "Online dictionary learning for sparse coding," in *Proc. of the 26th Annu. Int. Conf. on Mach. Learning*. ACM, 2009, pp. 689–696.

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

[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[6] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[7] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," *J. Machine Learning Research*, vol. 10, no. 1-41, pp. 66–71, 2009.

[8] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2004.

[9] S. Karygianni and P. Frossard, "Linear manifold approximation based on differences of tangents," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2011, pp. 973–976.

[10] S. Karygianni and P. Frossard, "Tangent-based manifold approximation with locally linear models," *Signal Processing*, vol. 104, pp. 232 – 247, 2014.

[11] W. K. Allard, G. Chen, and M. Maggioni, "Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis," *Appl. and Computational Harmonic Anal.*, vol. 32, no. 3, pp. 435–462, 2012.

[12] T. Ahmed and W. U. Bajwa, "A greedy, adaptive approach to learning geometry of nonlinear manifolds," in *Proc. IEEE Workshop on Statistical Signal Process.*, June 2014.

[13] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, 2010.

[14] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Int. J. Computer Vision*, vol. 100, no. 3, pp. 217–240, 2012.

[15] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Proc. Adv. Neural Inform. Process. Syst.*, 2002, pp. 681–688.

[16] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM review*, vol. 38, no. 3, pp. 367–426, 1996.

[17] P. L. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Eng.*, pp. 185–212. Springer, 2011.

[18] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert spaces," in *Advances in Order Restricted Statistical Inference*, pp. 28–47. Springer, 1986.

[19] H. H. Bauschke and P. L. Combettes, "A Dykstra-like algorithm for two monotone operators," *Pacific J. of Optimization*, vol. 4, no. 3, pp. 383–391, 2008.

[20] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Conf. Rec. of The Twenty-Seventh Asilomar Conf. on Signals, Syst. and Comput.* IEEE, 1993, pp. 40–44.

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

[22] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Proc. IEEE 12th Int. Conf. Comput. Vision Workshops*, 2009, pp. 234–241.