A generic unfolding algorithm for manifolds estimated by local linear approximations

Jonas Nordhaug Myhre[1], Matineh Shaker[2], Mustafa Devrim Kaba[3], Robert Jenssen[1], Deniz Erdogmus[4]

[1] UiT the Arctic University of Norway, [2] Microsoft AI & Research, [3] Johns Hopkins University,
[4] Northeastern University

Abstract

The individual stages of most popular manifold learning algorithms are complicated by overlapping ideas – often consisting of a mix of learning how to embed, unfold and reduce the dimension of the manifold at the same time. Furthermore, the effect each step has on the final result is in many cases not clear. Research in both machine learning and mathematical communities has focused on the steps involved in manifold embedding and estimation, and sample sizes and performance bounds related to these operations have been explored. However, the problem of unwrapping or unfolding manifolds has received relatively little attention despite being an integral part of manifold learning in general. In this work, we present a new generic algorithm for unfolding manifolds that have been estimated by local linear approximations. Our algorithm is a combination of ideas from principal curves and density ridge estimation and tools from classical differential geometry. Numerical experiments on both real and synthetic data sets illustrates the merit of our proposed algorithm.

1. Introduction

Manifold learning is one of the fundamental directions of Machine Learning research [42, 38, 45, 36, 44, 8, 35]. It is motivated by the notion that high dimensional data sets often exhibit intrinsic structure that is concentrated on or near manifolds of lower local dimensionality. This notion has inspired a plethora of algorithms such as ISOMAP [38], Local Linear Embedding [33], or Maximum Variance Unfolding [44].

These pioneering algorithms have major drawbacks. First, the steps of actually learning from manifolds – dimensionality reduction, manifold estimation and unfolding/linearizing the manifold – are all interlaced. Second, most algorithms fail to produce an actual manifold that is close to the true manifold the data are sampled from [6].

Recent research includes several works that separately analyze, and justify theoretically, the different steps of a general manifold learning framework. [16], [28] and [18] investigates and proves the *manifold estimation* part of the manifold learning framework. [30], [1] and [2] proposes practical algorithms for estimating manifolds. [43] defined a general and theoretically valid embedding algorithm for n-dimensional manifolds.

The final piece that has been less studied separately is the notion of unfolding. This is the key contribution of this paper: *A novel generic algorithm for unfolding manifolds estimated by a collection of local linear approximations*.

It is a well-known fact that manifolds that are isometric to $\mathbb{R}^d$ can always be unfolded into a linear subspace of $\mathbb{R}^d$ [25][1]. On the other hand, depending on the shape of the manifold (a function of both the reach of the manifold and the total volume of the manifold [28]), the trivial unfolding via a linear projection such as PCA is seldom likely to provide a proper isometric unfolding.

Instead, given a collection of local linear approximations as in [28] or a principal manifold estimate as in [30], we use the property that geodesics along a manifold isometric to $\mathbb{R}^d$ will neither converge or spread out [25]. This allows us to unfold the manifold directly by (1) transporting all local linear approximations to a reference point on the manifold via parallel transport along geodesics in the ambient space and (2) sending them back along the same geodesics, with the translation vector projected onto the local tangent space at each step. The last step is equivalent to setting to zero the last $D - d$ elements of the parallel translation vector, and ensures a d-dimensional global chart for the isometric manifold. We consider a global isometric chart in $\mathbb{R}^d$, representing the manifold $M \in \mathbb{R}^D$ as an unfolded version of the manifold.

To conclude this introduction, we acknowledge the fact that the isometry constraint on M is restrictive, but this

[1] The unfolding will always exist, but might be hard to implement in practice.

is a contribution towards a general understanding of manifold learning research. Also for example in analyzing latent spaces of (deep) generative models, a popular direction in modern manifold learning, it could be assumed that certain parts, e.g. different classes, of the latent space is isometric to $\mathbb{R}$.

Structure of paper: We begin with a short introduction to the relevant background theory in Section 2, namely *principal manifold estimation* and some basic differential geometry. We then proceed by presenting our algorithm and the approximations we introduce in Section 3. Section 4 and Section 5 presents numerical experiments and a short conclusion respectively.

2. Background material

This section starts with a brief review of Riemannian geometry. For a more complete overview, we refer the reader to the books of Tu, Lee or do Carmo [40, 25, 26, 13].

2.1. Differential geometry – a quick reminder

A *manifold* is a second countable, locally Euclidean Hausdorff space. Throughout this paper we are referring to *submanifolds* of $\mathbb{R}^D$. Given a manifold M diffeomorphic to $\mathbb{R}^d$, at each point $p \in M$ the *tangent space*, T_pM, is the Euclidean space of dimension d which is tangent to M at p [25]. The *tangent bundle* is the disjoint union of all tangent spaces of M. A *Riemannian manifold* is a smooth manifold equipped with a smoothly changing *metric*, given by the inner product $g : T_pM \times T_pM \to \mathbb{R}$. This metric defines lengths of vectors and curves on the manifold as well as the volume element[31]. The mapping from the tangent space T_pM to the manifold M is known as the *exponential map*, and the inverse transformation from M to T_PM as the *log map* [7]. Vectors in T_pM can be expressed by *local coordinates*, $x = (x_1, x_2, \dots, x_d)$ with an induced basis of differentials $E_i = \frac{\partial p}{\partial x_i}$ [25]. A *geodesic* is a locally minimizing curve along M between two points on M. *Parallel transport* is the translation of a tangent vector along M that keeps the length and direction of the vector[22, 34]. Finally we assume, for simplicity, throughout this paper that M is geodesically convex and does not contain holes. The latter can be relaxed and is covered by Rosman et al. [32], but this is beyond the scope of this work.

2.2. Manifold estimation – density ridges and local PCA

There are two types of manifold estimates relevant to this paper, *principal manifold estimation via density ridges* (PME) and manifold estimation via local PCA. This is also reflected in the recent theoretical contributions of [16], [28] and [17], where both local PCA and PME are shown to be useful empirical estimators that are close to the true manifold in Hausdorff distance.

Local PCA manifold estimation consists of selecting a collection of points with a corresponding local neighborhood, and then for each local neighborhood perform a PCA projection to d dimensions [28]. Thus the manifold estimate would correspond to $M = \bigcup_{i=1}^{K} C_i$, where C_i are the local coordinates obtained by local PCA and K is the number of local approximations used.

PME is based on *principal curves* that were originally introduced by Hastie and Stuetzle [20]. Several extensions were made, [23, 15, 39], until [30] redefined principal manifolds as being the *ridges* of a probability density estimate, [30].

Given a probability density $f(x)$, its gradient $g(x) = \nabla^T f(x)$ and Hessian matrix $H(x) = \nabla\nabla^T f(x)$, the ridge/PME can be defined in terms of the eigendecomposition of the Hessian matrix:

Definition 2.1 (Ozertem 2011)**.** A point x is on the d-dimensional ridge, R, of its probability density function, when the gradient $g(x)$ is orthogonal to at least $D-d$ eigenvectors of $H(x)$ and the corresponding $D - d$ eigenvalues are all negative.

Furthermore, they also propose an algorithm for estimating the ridges through orthogonal projections of points onto the principal manifold estimate. Given the spectral decomposition of H as $H(x) = Q(x)\Lambda(x)Q(x)^T$, where $Q(x)$ is the matrix of eigenvectors sorted according to the size of the eigenvalue. $\Lambda_{ii}(x) = \lambda_i$, $\lambda_1(x) > \lambda_2(x) > \dots$, is a diagonal matrix of sorted eigenvalues. Furthermore $Q(x)$ can be decomposed into $\left[Q_\perp(x)\; Q_\parallel(x)\right]$, where $Q_\perp$ is the d first eigenvectors of $Q(x)$, and $Q_\parallel$ are the $D - d$ smallest – sorted according to eigenvalue. The set $Q_\perp$ is referred to as the *orthogonal subspace* due to the fact that when at a ridge point, all eigenvectors in $Q_\perp$ will be orthogonal to $g(x)$. This yields the following initial value problem for projecting points onto a density ridge:

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = V_t V_t^T g(y_t), \tag{1}$$

where $V_t = Q_\perp(x(t))$ at $y_t = y(t)$, and $y(0) = x$. We denote the set of y's that satisfy equation (1), as the d-dimensional *principal manifold estimate*[2] $\hat{R}$. In practice, $\hat{f}(x)$ is usually estimated by a kernel density estimate or a gaussian mixture model.

Genovese et al. [19] showed that the kernel density ridges $\hat{R}$ are consistent estimators of the true underlying ridges R under Hausdorff loss (see Theorem 7 in their exposition).

[2]The $\hat{R}$ is inherited from the name *ridge*.

2.3. Some connections between principal manifold estimation and local PCA

To end this section on background material, we would like to address some properties connecting the PME and the local PCA estimate. By this we would like to answer the question of why the principal manifold estimation framework is introduced, when we claim that the methodology presented in Section 3 should hold for any locally linear manifold estimate? The answer lies in several such properties, of which several will be used directly later in this paper:

1. By construction the modes of $\hat{f}$, $\left\{m_i : \nabla^T f = 0\right\}$ are points which lie on the principal manifold estimate. These points can be found by the mean shift algorithm, [12]. This is also further backed up by Morse theory, [22], which states that a given a function defined on M, M itself is the union of the *unstable manifold* of the critical points of this function. Taking $\hat{f}$ as this function, with several possible critical points, these *unstable manifolds* corresponds to the gradient ascent trajectories of the mean shift algorithm [10, 3]. Thus, the attraction basins of $\hat{f}$ *and serve as good local neighborhoods for approximation by local PCA*

2. A single local PCA estimate serves as an estimate of T_pM at a certain selected point on M. Conversely, given a principal manifold estimate $\hat{R}$ and a point $p \in \hat{R}$, $Q_{\|}(p)$ is a basis for $T_p\hat{R}$, see e.g. [19]. Moreover, if we look closer at the expression for the Hessian of $\hat{f}$ at x,

$$H(x) = \frac{1}{n}\sum_{i=1}^{n}\left(u_i u_i^T - \frac{1}{\sigma^2}I\right)K\left(\frac{||x-x_i||}{\sigma^2}\right), \tag{2}$$

where $u_i = (x - x_i)$, we see that the Hessian matrix is a local approximation – due to the restriction of points included by $K(\frac{||x-x_i||}{\sigma^2})$– of the covariance matrix used in PCA.

Thus, a manifold estimate based on the critical points of $\hat{f}$ and the projection of their corresponding basins of attraction onto $Q_{\|}(m_i)$ is an intuitive, theoretically justified and computationally efficient – no need to solve (1) for all points – compromise between PME and Local PCA. In light of this, we use this combined with kernel density estimation throughout the rest of this paper.

3. A generic algorithm for isometric unfolding

In this section we aim to answer the question: how can we unfold manifold estimates consisting of local linear approximations? We will answer this question by proposing a simple algorithm founded in basic geometric properties. It is assumed that the manifold estimate is of good quality, and we refer the reader to [16] and [28] for examples of error quantification and algorithms.

We start by outlining the main steps of our algorithm in Algorithm 1, and then proceed by explaining underlying principles behind our unfolding scheme. Initially, due to the assumption of M being isometric[3] to $\mathbb{R}^d$, we know that the manifold can be covered by a single global chart $\subset \mathbb{R}^d$. Furthermore, this assumptions also ensures that the length of geodesics along the embedded manifold should be preserved in this global chart. Given a sufficiently accurate manifold estimate, such as by a principal manifold estimate or local PCA, we already have a collection of local linear charts for the manifold (we note that our definition of *chart* is a slight abuse of terminology, but we follow the definitions of [31] where coordinates defined on a local tangent space can be considered an approximation of a chart on the manifold). Thus, what is needed to obtain a globally unfolded chart is to orient all local charts according to the direction and length of the geodesics connecting them. Here, we note the similarity of our methodology to that of *Fast manifold learning by Riemannian normal coordinates* [7], where a global chart is directly created by calculating the geodesic distance and initial angle to *all* data points and keeping only the length and initial angle information.

Algorithm 1 Generic unfolding algorithm

Input: A collection of local linear approximations (charts)

1: Select a reference point of unfolding.
2: Calculate geodesics from the reference point to the origin of all local linear approximations.
3: Transport all local charts along the corresponding geodesic towards the reference point.
4: Transport all local charts back along the geodesic, setting the last $D-d$ coordinates of the parallell transport vector to zero to enable d dimensional transport.

Output: Global unfolding.

To be more concrete, we first select a reference point p on the manifold we wish to unfold. We recall that each chart consists of a local linear approximation as well as a point of origin on the manifold. Given that the manifold is geodesically convex, p can be connected to all other charts via geodesics. Moreover, this allows us to apply *parallel transport* and translate all charts to the same reference point. This will center all charts at p, and by projection

[3]This is a necessary assumption if we expect a proper unfolding of the manifold – most often this is overlooked in manifold learning methodology.

onto T_pM we get at d dimensional chart over M. To further emphasize our point, this is illustrated in (a) to (d) of Figure 1.

Finally to carry out the actual unfolding (a global chart in $\mathbb{R}^d$), the last $D-d$ coordinates of the parallel transport vector will be set to zero and d dimensional transport (translation constrained to stay in the global chart) is obtained. This is illustrated in (d) to (f) of Figure 1. Since M is isometric to $\mathbb{R}^d$ the length of the geodesic is trivially preserved. This also ensures the orientation of the charts to be preserved, as the isometry condition implies zero intrinsic curvature – the geodesics will not diverge or converge along their original path [25]. In practice, due to ambiguity in the orientation of local charts and the directions of geodesics, the actual implementation is somewhat more involved. The complete algorithm is presented in Appendix A.

We end this section with an illustration of the unfolding procedure in practice. Figure 1 shows the, by now famous, swiss roll dataset unfolded by our algorithm. Each subfigure shows a different stage of the procedure and in Figure 2 the entire unfolding is shown. The next two sections presents necessary details on the approximation and practical implementations of our algorithm; *geodesics* and *parallel transport*.

3.1. Approximating geodesics

Given a metric tensor at each point of the manifold solving the Euler Lagrange equation admits a computationally tractable scheme for finding geodesics through solving a system of differential equation [21] . As the PME framework does not explicitly provide a metric tensor, we reframe an idea presented by Dollár et al. [14]. The idea is to minimize the distance between two points using gradient descent, while at the same time keeping the starting points and endpoints fixed and making sure that all points on the path lies approximately on the manifold.

Given a sequence of points $\{\gamma_i\}_{i=1}^n \in M$ between two points on the manifold x and y we can formulate the problem of finding a shortest path constrained to the manifold as follows:

$$\begin{aligned} \underset{\gamma}{\text{minimize}} \quad & \sum_{l=2}^{n-1} ||\gamma_l - \gamma_{l-1}||^2 \\ \text{subject to} \quad & \gamma_1 = x,\ \gamma_n = y,\ \gamma \in M. \end{aligned} \tag{3}$$

To optimize (3) the path γ is initialized using Dijkstra's algorithm and further discretized with linear interpolation between the n given points. Then minimization is performed by alternating between gradient descent to shorten distance and projection to the PME, equation (1), to ensure that points stay on the manifold.

In addition we use another idea from the work of [14] for fast out of sample projections. After a selection of points have been projected to the PME, by equation (1), the tangent space of the ridge estimator is at each point x spanned by the parallel Hessian eigenvectors $Q_{\|}(x)$. To project a new out-of-sample point x_o orthogonally to the manifold estimate, $||x_o - x_M||^2$, where x_M is a point on the ridge, needs to be minimized. This can be solved by setting x_M to the closest point of x_o on the manifold and then iterating over $x_M \leftarrow x_M + \alpha Q_{\|}(x_M) Q_{\|}(x_M)^T (x_o - x_M)$.

3.2. Approximate parallel transport

An alternative definition of a geodesic is that it can be characterized as a curve with a velocity vector field that is parallel to the curve [25]. This enables the translation of vector fields, e.g. the local linear approximations at T_pM, along a geodesic.

Given C_i, the local coordinates of the manifold estimate centered at some point $m_i \in M$, let $\boldsymbol{\gamma}$ be the vector containing the points of the approximate geodesic from $\gamma_0 = m_i$ to a point $\gamma_n = m_r$ – e.g. the reference point – as found by solving (3). The approximate parallel transport is performed by translating the local coordinates C_i along the finite difference tangent vectors of the approximate geodesic γ. At each step of the translation the points are projected to the local tangent space by $Q_{\|}(\gamma_t) Q_{\|}(\gamma_t)^T$. This is to ensure that the local coordinates stay in the tangent space all the way along the geodesic towards the target point m_r. The algorithm for approximate parallel transport is summarized in Algorithm 2.

Algorithm 2 Approximate parallel transport from m_i to m_j

Input: Coordinate vectors C_i and approximate geodesic $\boldsymbol{\gamma} = [\gamma_t]_{t=1}^n$.

Calculate $Q_{||}(\gamma_t)$ for $t = [1, \cdots, n]$.
Initialize $C_i' = C_i$.
for $t = [2, \cdots, n]$ **do**
    $C_i'' = Q_{||}(\gamma_t) Q_{||}(\gamma_t)^T \left(C_i' + (\gamma_t - \gamma_{t-1}) \right)$
    $C_i' = C_i''$
end for

Output: Local coordinates C_i' transported from m_i to m_j along the manifold.

4. Experiments

In this section, we present empirical results illustrating our generic unfolding framework on both real and synthetic data sets. Our intention is to illustrate prototypical unfolding scenarios and that our algorithm gives meaningful results. Each experiment is designed to illustrate different aspects of task of unfolding manifolds. In addition, we also include an illustration of how the unfolding algorithm be-

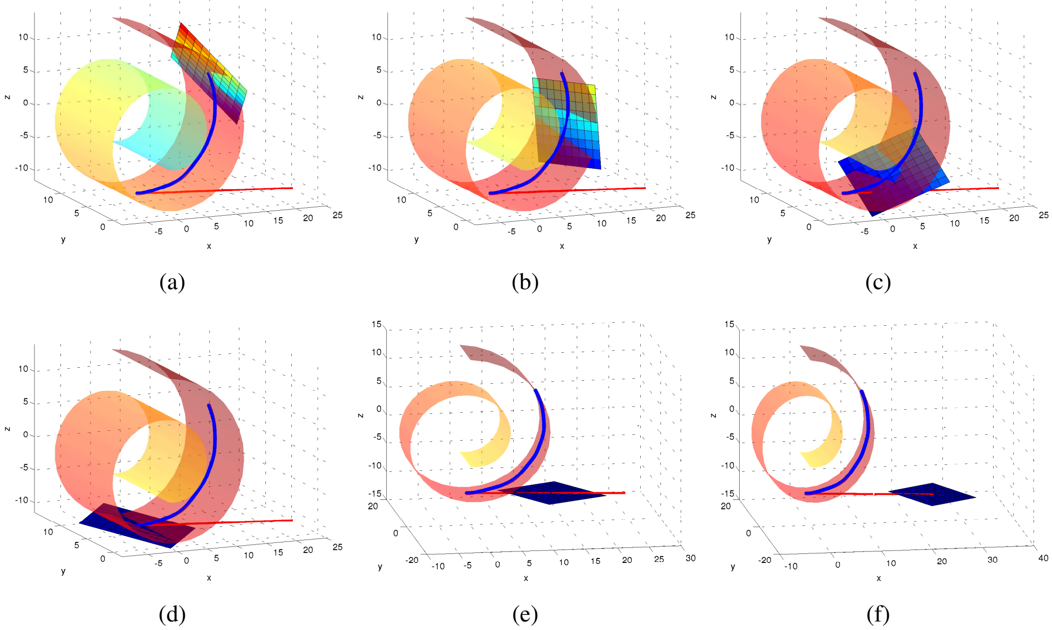


Figure 1: These figures illustrate the main intuition behind our unfolding strategy: The point where the red and blue lines meet is the reference point of unfolding. (a) to (d) illustrates the parallel translation of a local linear approximation (shown as a linear subspace) to the reference point along the geodesic (shown in blue). (d) to (f) shows the final unfolding steps, where the translation back along the geodesic has been restricted to $d = 2$. The final unfolding result, including all local linear approximations, is shown in Figure 2. This figure is best viewed online or in color.

haves when we try to unfold a manifold (a hemisphere) that is *not isometric* to $\mathbb{R}^d$. For all the data sets we use the average distance to the 5th neighbor heuristic rule for the bandwidth of the kernel density estimate [29] and all data sets have been standardized to have zero mean and unit variance.

4.1. Swiss roll dataset

We begin by testing our algorithm on the so-called swiss roll dataset [42, 33]. It consists of a deformed but isometric two-dimensional manifold embedded in $\mathbb{R}^3$. We use the mean shift algorithm to obtain a set of reference points on the manifold and we perform a linear projection of all points belonging to each reference point. Recall that mean shift provides a partitioning of the manifold according to Morse theory [22, 10]. We select the point with the lowest z coordinate as reference point of unfolding and then proceed with all steps mentioned in Algorithm 1. The result is shown in Figure 2. We see from the figure that the local relationships of the linear approximations, marked in color, are preserved.

4.2. MNIST handwritten digits

We test our algorithm on the ones of the MNIST handwritten digits dataset [24]. Due to the deterioration of the performance of the kernel density estimator as a function of dimension, we pre-process the data set by performing a global dimensionality reduction to 10 dimensions using PCA. The dimensionality of MNIST has previously been found to be approximately 10 [27, 9, 37]. We apply the same steps as for the swiss roll data set. Due to the restricted variations of the digit one, we make the assumption that the underlying manifold is isometric to $\mathbb{R}^2$ and run our algorithm. The result is shown in Figure 3, with a uniform selection of the original images overlain on top of the unfolded chart. We clearly see that both the orientation and

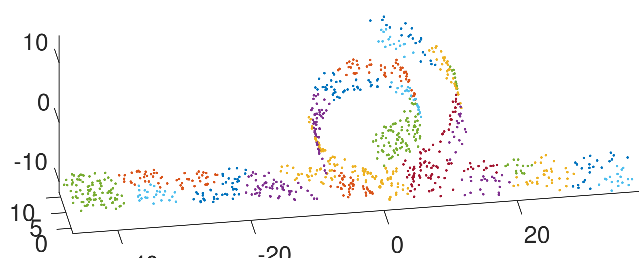


Figure 2: Swiss roll data set. Both the original data set and the unfolded version are shown. The reference point is the lowest mean shift mode which(also seen in Figure 1).

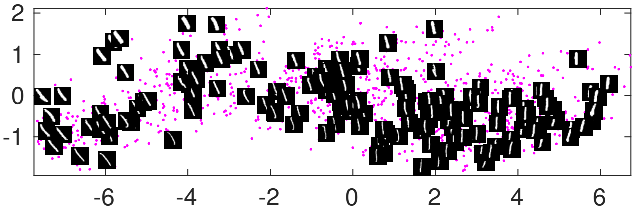


Figure 3: Unfolded version of MNIST ones shown as a single chart in $\mathbb{R}^2$.

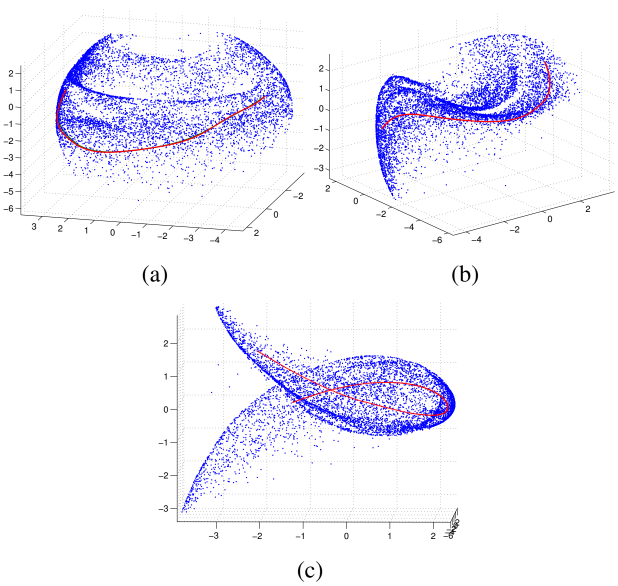


Figure 4: Principal manifold estimate (blue points) and an example of an approximate geodesic (red) between two random points of the top 10 principal components of the MNIST ones. **(a)** PCA dimensions 1 to 3. **(b)** PCA dimensions 1, 2, and 4. **(c)** PCA dimensions 2 to 4.

thickness of the one digits is captured in each dimension of the chart. In Figure 4 we see an example of the approximate geodesic used in the MNIST unfolding, the results are shown on the top three, 1, 2, 4 and 2, 3, 4 principal components respectively. We clearly see that even though the manifold is nonlinear in the different dimensions, the approximate geodesic is able to smoothly follow the manifold estimate.

4.3. Comparison with alternative algorithms

We compare our algorithm to a selection of benchmark manifold learning algorithms. For clarity, we use a very simple toy data example consisting of a small part of the swiss roll dataset. Furthermore, to illustrate the smoothing properties of the principal manifold estimate, we add isotropic Gaussian noise with the same dimension as the ambient space. The goal of this experiment is to illustrate that our algorithm gives meaningful results when the main goal is restricted to *unfolding*. The data set with and without noise and the unfolded version using our algorithm is shown in Figure 5. In Figure 6 we see the data processed to two-dimensions by Isomap, LLE, MVU, LTSA, Laplacian eigenmaps and our algorithm [38, 33, 45, 44, 5]. The true underlying chart is shown with red dots, and the unwrapped coordinates from the different algorithms are shown with the same color coding as in Figure 5. All result were obtained with neighborhood parameter $k = 12$ and both the results and true underlying parameterization have been normalized and centered.

Looking at the figures we see that Isomap is the closest to ours, but it is not able to capture the manifold structure without retaining the noise. The other four algorithms, Laplacian eigenmaps, Local linear embedding, Maximum variance unfolding and Local tangent space aligment all fail to unfold the underlying manifold and the results are accordingly hard to comment on. We tried different parameters for all algorithms, but all failed to give reasonable results, except Isomap and our algorithm. Due to this we choose to show only the results of the default parameter $k = 12$.

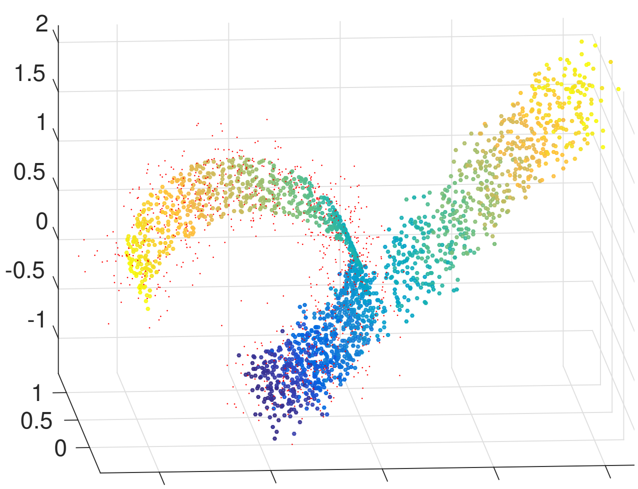
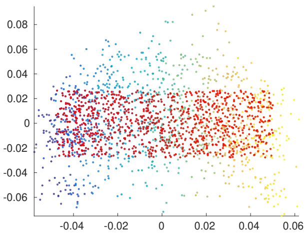


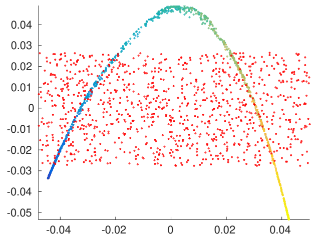
Figure 5: Dataset with (small red dots) and without (large colored dots) noise. The result of our algorithm projected to the tangent space of a reference mode shown in the same colors as the data without noise.

4.4. What happens when the manifold is not isometric to $\mathbb{R}^d$?

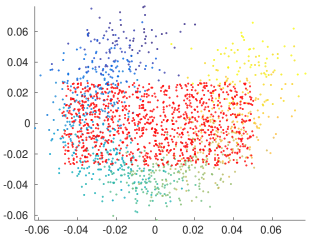
As a final experiment we illustrate how our algorithm behaves when the manifold we want to unfold is *not isometric* to $\mathbb{R}^d$. We illustrate this on a hemisphere toy data set. In general a sphere has constant positive curvature and



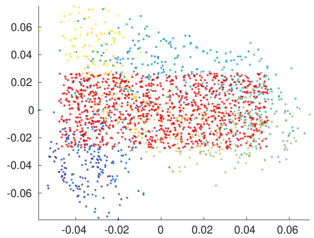
(a) Isomap $k = 12$.



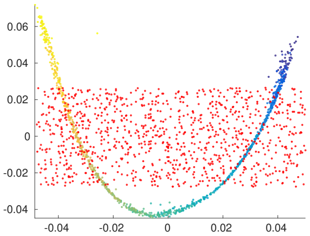
(b) Laplacian eigenmaps $k = 12$.



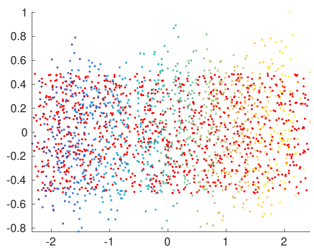
(c) Local linear embedding $k = 12$.



(d) Maximum variance unfolding $k = 12$.

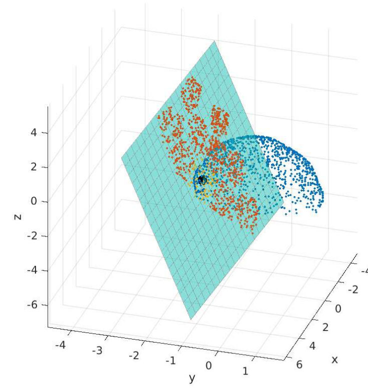


(e) Local tangent space alignment $k = 12$.

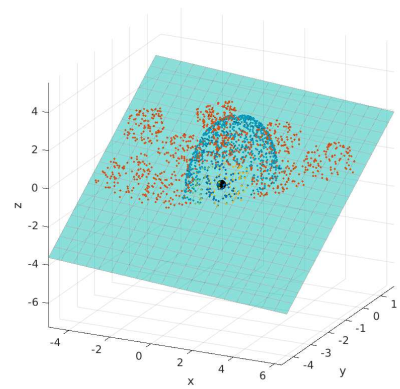


(f) Our algorithm

Figure 6: Uncovering the underlying parameterization of the manifold shown in Figure 5 using various benchmark manifold learning algorithms. The small red dots shows the true underlying parameterization, while the colored dots shows the output from the various manifold learning algorithms.



(a)



(b)

Figure 7: An example where the algorithm fails. The red dots represents the unfolded local linear approximations of the manifold. The black dot is the reference point. The tangent space of the reference point is shown as a shaded plane. This figure is best viewed online.

is therefore not isometric to $\mathbb{R}^d$. Unfolding this set without shearing or breaking is not possible. In Figure 7 we see that the algorithm breaks the manifold when unfolding it. Even though this might be considered a negative result, this is in accordance with the theory and we see that the algorithm keeps the structure of the local tangent vectors. Furthermore, seen relative to the reference point, the euclidean distance of the unfolded structure is close to the geodesic distance. On the other hand, it is obvious that relative distances across local approximations not connected directly to the reference point gets broken.

5. Conclusion

In this paper we have illustrated the principles and proposed a methodology needed to create a generic unfolding strategy for principal manifold and local PCA estimates. We have shown simple numerical experiments that back up our claims, which are all motivated by well established mathematical theory.

There are several steps needed to perform the unfolding in this methodology, most notably density estimation, calculating geodesics and performing parallel translation. All of these steps carry with them some form of inaccuracies which needs to be further studied and quantified. [16, 28, 19, 11] have all come a long way in analysing manifold estimation from noisy data, and some of their techniques could be extended to our work.

The kernel density estimation part of the framework clearly presents a weakness, especially when faced with very high-dimensional input spaces. But due to the generality of the methodology, this should not have major impact in further practical applications.

Further work in this direction would most certainly involve analysis of latent spaces of Deep Generative Models. Several works are already investigating the geometry of such latent spaces using geometry and manifold learning [34, 4, 41].

A. Algorithm for unfolding transported coordinates

The algorithm for unfolding the transported tangent space vectors is presented in Algorithm 3. The operator null() returns the null space of a matrix, and $\Delta\nu = \nu_t - \nu_{t-1}$ is the difference vector between two points along a curve.

Algorithm 3 Isometric unfolding

Input: Local coordinates transported to a reference mode $C = \{C_i^{'}\}_{i=1}^{m-1}$ and the geodesic from the reference mode to all other modes, ν_i.

Choose a basis E_i for the tangent space of a point along the geodesic ν_t, $T_{\nu_t}M$:

$$E_i = \begin{bmatrix} \Delta\nu_t & \text{null}\left(\begin{bmatrix} \Delta\nu_t & Q_\perp(\nu_t) \end{bmatrix}\right) \end{bmatrix}.$$

for all Translated charts $C_i^{'}$ **do**

  for $t = [2, \cdots, n]$, n is the number of steps in $\boldsymbol{\nu}_i$ **do**

    Choose the translation direction as being along ν_t such that the step along the geodesic will be $\xi = [\|\nu_t'\|, \mathbf{0}]^T, \xi \in \mathbb{R}^D, \mathbf{0} \in \mathbb{R}^{D-d}$.

    Ensure that the orientation of the basis vectors stay consistent: $B = BE_t^T E_{t+1}$, initialized as $B = E_1$.

    Translate and project to the next tangent space along the geodesic,

$$U_i = Q_\parallel(\nu_t) Q_\parallel(\nu_t)^T \left(C_i^{'} + B\xi_t\right)$$

  end for

end for

Output: Unfolded manifold consisting of all charts $\widehat{M} = \bigcup_{i=1}^{m} U_i$.

References

[1] Talal Ahmed and Waheed U Bajwa. Geometrie manifold approximation using union of tangent patches. In *GlobalSIP*, pages 458–462, 2014.

[2] William K Allard, Guangliang Chen, and Mauro Maggioni. Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis*, 32(3):435–462, 2012.

[3] Ery Arias-Castro, David Mason, and Bruno Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *The Journal of Machine Learning Research*, 17(1):1487–1514, 2016.

[4] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

[5] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[6] Jean-Daniel Boissonnat, Leonidas J Guibas, and Steve Y Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. *Discrete & Computational Geometry*, 42(1):37–70, 2009.

[7] Anders Brun, Carl-Fredrik Westin, Magnus Herberthson, and Hans Knutsson. Fast manifold learning based on riemannian normal coordinates. In *Image Analysis*, pages 920–929. Springer, 2005.

[8] Christopher JC Burges. Dimension reduction: A guided tour. *Machine Learning*, 2(4):275–365, 2009.

[9] Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern recognition*, 47(8):2569–2581, 2014.

[10] José E Chacón. Clusters and water flows: a novel approach to modal clustering through morse theory. *arXiv preprint arXiv:1212.1384*, 2012.

[11] Yen-Chi Chen, Christopher R Genovese, and Larry Wasserman. Asymptotic theory for density ridges. *arXiv preprint arXiv:1406.5663*, 2014.

[12] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[13] Manfredo P Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016.

[14] Piotr Dollár, Vincent Rabaud, and Serge Belongie. Non-isometric manifold learning: Analysis and an algorithm. In *Proceedings of the 24th international conference on Machine learning*, pages 241–248. ACM, 2007.

[15] Jochen Einbeck, Gerhard Tutz, and Ludger Evers. Local principal curves. (1992):301–313, 2005.

[16] Charles Fefferman, Sergei Ivanov, Yaroslav Kurylev, Matti Lassas, and Hariharan Narayanan. Fitting a putative manifold to noisy data. In *Conference On Learning Theory*, pages 688–720, 2018.

[17] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[18] Christopher R Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. The geometry of nonpara-

metric filament estimation. *Journal of the American Statistical Association*, 107(498):788–799, 2012.

[19] Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Nonparametric ridge estimation. *The Annals of Statistics*, 42(4):1511–1545, Aug. 2014.

[20] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[21] Søren Hauberg, Oren Freifeld, and Michael J Black. A geometric take on metric learning. In *Advances in Neural Information Processing Systems*, pages 2024–2032, 2012.

[22] Jürgen Jost. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008.

[23] Balázs Kégl, Adam Krzyzak, Tamás Linder, and Kenneth Zeger. Learning and design of principal curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(3):281–297, 2000.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer, 1997.

[26] John M Lee. *Smooth manifolds*. Springer, 2003.

[27] Gabriele Lombardi, Alessandro Rozza, Claudio Ceruti, Elena Casiraghi, and Paola Campadelli. Minimum neighbor distance estimators of intrinsic dimension. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 374–389. Springer, 2011.

[28] Kitty Mohammed and Hariharan Narayanan. Manifold learning using kernel density estimation and local principal components analysis. *arXiv preprint arXiv:1709.03615*, 2017.

[29] Jonas Nordhaug Myhre and Robert Jenssen. Mixture weight influence on kernel entropy component analysis and semi-supervised learning using the lasso. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.

[30] Umut Ozertem and Deniz Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12(4):1249–1286, 2011.

[31] Dominique C Perrault-Joncas, Marina Meilă, and James McQueen. Improved graph laplacian via geometric consistency. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4460–4469. Curran Associates Inc., 2017.

[32] Guy Rosman, Michael M Bronstein, Alexander M Bronstein, and Ron Kimmel. Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89(1):56–68, 2010.

[33] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[34] Hang Shao, Abhishek Kumar, and P Thomas Fletcher. The riemannian geometry of deep generative models. *arXiv preprint arXiv:1711.08014*, 2017.

[35] Carlos Oscar Sánchez Sorzano, J Vargas, and A Pascual Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014.

[36] Ke Sun, Eric Bruno, and Stéphane Marchand-Maillet. Stochastic unfolding. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6. IEEE, 2012.

[37] Ke Sun and Stéphane Marchand-Maillet. An information geometry of statistical manifold learning. In *ICML*, pages 1–9, 2014.

[38] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[39] Robert Tibshirani. Principal curves revisited. *Statistics and Computing*, 2(4):183–190, 1992.

[40] Loring W Tu. *An introduction to manifolds*, volume 200. Springer, 2008.

[41] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7064–7073, 2017.

[42] Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[43] Nakul Verma. Distance preserving embeddings for general n-dimensional manifolds. *The Journal of Machine Learning Research*, 14(1):2415–2448, 2013.

[44] Kilian Q Weinberger and Lawrence K Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, volume 6, pages 1683–1686, 2006.

[45] Zhen-yue Zhang and Hong-yuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University (English Edition)*, 8(4):406–424, 2004.