IsUMap: Manifold Learning and Data Visualization leveraging Vietoris-Rips Filtrations

Parvaneh Joharinad [*1,2], **Hannaneh Fahimi** [*1,2], **Lukas Silvester Barth** [*2], **Janis Keck** [*2, 3], **Jürgen Jost**[2]

[1] Center for Scalable Data Analytics and Artificial Intelligence (ScaDS,AI) Dresden/Leipzig, Germany,
[2] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany,
[3] Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany
parvaneh.joharinad@mis.mpg.de, fatemeh.fahimi@mis.mpg.de, lukas.barth@mis.mpg.de, janis.keck@maxplanckschools.de, jjost@mis.mpg.de

Abstract

We introduce IsUMap, a novel manifold learning technique. It enhances data representation by integrating aspects of UMAP and Isomap with Vietoris-Rips filtrations and metric realization of one-parameter filtrations of simplicial complexes.

Inferring topological information from combinatorial models which have been built according to metric relations (Vietoris-Rips complexes) has proven useful in topological data analysis and general machine learning applications. This encourages the use of such objects for geometric inference.

We extend this research direction with a clear theoretical pipeline. It not only provides a comprehensive guide for assigning a (triangulated) metric space to every admissible one-parameter filtration of simplicial complexes but also offers a method for merging these objects. Our method thus presents a systematic and detailed construction of a metric representation for locally distorted metric spaces that captures complex data structures more accurately than previous schemes. It also addresses limitations in existing methods by accommodating non-uniform data distributions and intricate local geometries. We validate its performance through extensive experiments on examples with known geometries and in applications to data, in particular from computational biology.

Code and extended article with appendix available at — https://github.com/LUK4S-B/IsUMap

1 Introduction

A good representation of complex data may enable an intuitive understanding of important features, making it indispensable for data analysis in a wide range of scientific disciplines and applications. Good representations are often more compact (and hence computationally more efficient) than the original data, while capturing or even enhancing the relevant features. This makes them useful for downstream tasks, transfer learning and a better understanding of the data. The many tools for transforming complex data into low-dimensional representations include feature learning or representation learning techniques. These enable systems to automatically discover the necessary representations for various downstream tasks. A standard strategy in representation learning is to train a deep neural network, call one of the latent layers the representation layer and investigate it with a variety of techniques, cf. (Bengio, Courville, and Vincent 2013; Botteghi, Poel, and Brune 2022). While fascinating and fruitful, a disadvantage is that those representations are hard to interpret because of the large number of learnable parameters. In this article, we thus focus on a manifold learning algorithm that has only very few predefined parameters and is hence particularly transparent. Topological Data Analysis (TDA) translates the homology classes of a filtered simplicial complex of Vietoris-Rips (VR) type into a set of barcodes (Zomorodian and Carlsson 2004; Carlsson 2009) to capture the persistent topological features of a point cloud. Simplicial complexes can encode relations, e.g. in social networks, that are of higher-order than those in binary metric relations. This suggests them as natural mathematical objects for data analysis, even when the analysis is geometric, and extends beyond inferring topological characteristics as in TDA. Our approach therefore is also different from TDA. It leverages the VR filtration corresponding to the given metric, to combine local geometries and to achieve a unified global geometric representation. Thus, our new method, called *IsUMap* (operating in the category of uber-metric spaces (UM))), combines ideas from TDA with the established dimensionality reduction methods of UMAP (McInnes, Healy, and Melville 2018) and Isomap (Tenenbaum, Silva, and Langford 2000). IsUMap is specifically designed to address the issue of non-uniform data distribution on a low-dimensional manifold by applying local distortions to the distance function, thereby overcoming the limitation of the above mentioned approaches. But this leads to the challenge of aggregating pairwise similarities between data points, represented by a distance function, across overlapping neighborhoods with different geometries. When the data is initially sampled from a Riemannian manifold (a natural assumption), we eventually obtain a triangulated metric space composed of distorted local pieces of that manifold, but with a unified intrinsic global distance function. This

[*]These authors contributed equally.

can then be used for an embedding of the dataset into a low-dimensional space while preserving the global metric structure, e.g. by Multidimensional Scaling (MDS, c.f. (Torgerson 1952; Lee, Verleysen et al. 2007)) or metric MDS, c.f. (Abdi 2007).

1.1 Contributions

IsUMap integrates key concepts from UMAP and Isomap, inferring global topology and global intrinsic geometry respectively, and provides a low dimensional representation that preserves both of them, while uniformizing the data distribution if desired. This hybrid approach improves on the limitations of either method.

IsUMap is a theoretically grounded methodology, designed to facilitate the use of higher-order correlations. It efficiently reconstructs complex geometrical features in data, particularly in cases with non-uniform data distributions.

1.2 Related Works

Our method uses aspects of both UMAP (McInnes, Healy, and Melville 2018) and Isomap (Tenenbaum, Silva, and Langford 2000). UMAP is a dimensionality reduction method with the goal of uniformizing a metric and preserving *global topological* features of the data. Isomap, on the other hand, emphasizes preserving local geometric features while ensuring that these local details contribute to an accurate representation of the *global geometry*. IsUMap combines these aspects, enabling it to retain both topological and geometric information, thereby offering improvements over both methods.

As typical in manifold learning, we first construct a weighted graph to model the dataset, which is then optimized to align with a low dimensional embedding. While there are various strategies for assigning weights, such as making them proportional to pairwise distances as in classical methods like MDS (Torgerson 1952; Lee, Verleysen et al. 2007), or inversely related to the distance, defining a probability of edge appearance as in Laplacian Eigenmaps (LE) (Belkin and Niyogi 2003), IsUMap offers a flexible framework that facilitates transitions between these approaches. Similar to UMAP, its predecessor t-SNE (van der Maaten and Hinton 2008) optimizes a Kullback-Leibler divergence on the probability weight function over the edge set. However, t-SNE constrains the weight function to define a random walk, requiring iterated local normalization that may increase the computational cost considerably.

While pairwise information remains the popular approach due to computational constraints, some methods, such as TriMap (Amid and Warmuth 2019), have begun exploring the incorporation of higher-order relationships within datasets. The theory underlying IsUMap is designed to facilitate the use of higher-order correlations. However, the current implementation of our algorithm is limited to graph representations, leaving the computationally more costly integration of higher-dimensional simplices for the future. One reason for the success of our method lies in its foundation on the construction of Vietoris-Rips filtrations. These filtrations, built from geometric information, inherently encode topological details, despite being entirely determined by pairwise correlations.

2 Theory

2.1 Weighted and Fuzzy Simplicial Complexes

Simplicial complexes are higher order generalizations of graphs. A simplicial complex is a collection S of subsets of a vertex set X called simplices which is closed downward under inclusion, i.e. whenever a simplex $\sigma \in S$, then also all of its subsets (or faces) are in S as well. In TDA, these combinatorial objects are used to infer information about topological features. A weighted simplicial complex carries a weight function $W : \sigma \mapsto W(\sigma) \in [0, \infty]$, whose the interpretation depends on the application. It is natural to assume that the weights are monotone with respect to taking subsets of simplices, $\sigma \subset \sigma' \implies W(\sigma) \leq W(\sigma')$. But we want to interpret the weights as membership strengths (or probabilities), and so, we instead restrict them to $[0, 1]$ and reverse the arrow of monotonicity ($\sigma \subset \sigma' \implies W(\sigma) \geq W(\sigma')$), and obtain a *fuzzy simplicial complex*. Of course, we can easily switch between the two versions. Let $\phi : [0, \infty] \to [0, 1]$ be a right-continuous, monotone decreasing function with $f(0) = 1, f(\infty) = 0$. Then a weighted simplicial complex (S, W) becomes a fuzzy simplicial complex $(S, \phi \circ W)$. We shall always use this inverse relationship between weights and probabilities in the sequel. In (McInnes, Healy, and Melville 2018), such fuzzy weights are merged by t-conorms (see below and Appendix A).

One-Parameter-Filtrations. One-parameter filtrations of complexes are families of simplicial complexes, indexed by a scalar parameter r, evolving as the parameter increases, i.e., $S_r \subseteq S_{r'}$ if $r < r'$. Each simplex 'appears' at a certain value of r and is then present for all higher values. A one-parameter filtration yields a weighted simplicial complex, with r (the instance the simplex appears in the filtration) as the weight, and conversely. Important examples are the Vietoris-Rips (VR) filtrations assigned to a metric space (X, d), as used in TDA for topological inference, and in manifold learning. Let (X, d) be a *uber*-metric space (a metric space where distances may be infinite). The VR-complex $VR(X, r)$ of (X, d) is the simplicial complex with vertex set X, where a finite subset $\{x_0, x_1, \ldots, x_n\}$ of X spans a simplex when its diameter is $\leq r$. It is computationally simple because it is completely determined by its 1-skeleton (the edges), thereby avoiding a combinatorial explosion in the higher order simplices (see appendix A for more on VR-complexes and their application in TDA). Although we choose VR-Complexes for their computational advantages, there exist more general combinatorial models , e.g. Čech-complexes, which are not solely determined by their edges.

2.2 Metric Realization of Weighted Simplicial Complexes

It can be easily checked whether a given fuzzy-simplicial complex comes from a VR-filtration.

For an unweighted simplicial complex S with a finite vertex set X, there is a canonical way to turn S into a metric space.

Ordering the vertices, we use the unit $(n-1)$-simplex in $\mathbb{R}^n$

$$\Delta^{n-1} := \left\{ (t_0, \dots, t_{n-1}) \in \mathbb{R}^n \middle| \sum_{i=0}^{n-1} t_i = 1 : t_i \geq 0 \right\} \tag{1}$$

and map each combinatorial simplex $\{x_{i_0}, x_{i_1}, \dots, x_{i_k}\}$ to the convex hull of the corresponding standard unit vectors $\{e_{i_0}, e_{i_1}, \dots, e_{i_k}\}$. This yields the standard geometric realization $|S|$ of S.

But when we use the metric from the Euclidean ambient space on this realization of $VR(X, r)$, in general, we do not preserve the geometry of X such as distances between pair of vertices. Topological properties, however, are preserved, as by the *Hausmann theorem* (Hausmann 1995; Latschev 2001) the realization is homotopic to X (under some technical conditions). But for preserving geometric information, we can use fuzzy simplicial complexes (Spivak 2009). We realize each $(k+1)-$simplex with weight a in $(S, \phi \circ W)$ by putting $\sum_{i=0}^{k} t_i = \phi^{-1}(a)$ instead of $= 1$ in (1).

For realization of the entire object, we can then use natural gluing operations along shared facets. Moreover, the geometric realization of the $1-$skeleton suffices for inferring the metric on the vertex set, given by

$$d(x, y) = \inf_{x_1 = x, \dots, x_n = y} \sum_{i=1}^{n-1} \phi^{-1}(w(x_i, x_{i+1})), \tag{2}$$

which is, after applying the translation function ϕ^{-1}, the geodesic graph distance.

2.3 Merging Fuzzy Complexes

Fuzzy sets may naturally be merged via a t-conorm, i.e., a binary operation $T^{co} : [0,1] \times [0,1] \to [0,1]$ that is commutative ($T^{co}(a,b) = T^{co}(b,a)$), monotonic ($T^{co}(a,b) \leq T^{co}(c,d)$ if $a \leq c$ and $b \leq d$), associative ($T^{co}(a, T^{co}(b,c)) = T^{co}(T^{co}(a,b), c)$), and has an identity element 0 ($T^{co}(a, 0) = a$). By associativity, the product $\prod_j^{T^{co}}(a_j)$ does not depend on the order of operations.

With a t-conorm T^{co} and a translation function ϕ, we can merge finitely many weighted graphs (X, W_j) on the same vertex set X into a single weighted graph. To obtain the geodesic distance, we convert each graph to a fuzzy graph, merge via a t-conorm, convert back to the weighted setting, and compute the distance by (2). This is natural from the perspective of category theory, c.f. (Barth et al. 2024). When the weights W_j are themselves metric and we use $T^{co}(a,b) = \max(a,b)$, the resulting metric is the well-known gluing metric of metric geometry, c.f. (Burago, Burago, and Ivanov 2001) and Appendix B.

2.4 Uniformization via Conformal Transformations

Data are typically given in some $\mathbb{R}^d$, but often assumed to be sampled from a smooth lower-dimensional manifold M, or at least approximately so. As a submanifold of $\mathbb{R}^d$, M carries a Riemannian metric, and it therefore is locally approximately Euclidean. The Riemannian metric g defines inner products on the tangent spaces $T_pM, p \in M$ and induces a distance function d_g on M where $d_g(x,y)$ is the infimum of the lengths $l(\gamma) = \int_0^1 \sqrt{g(\dot\gamma(t), \dot\gamma(t))}dt$ of curves γ connecting them. We can then try to represent the data in a lower dimensional Euclidean space by minimizing some loss function comparing the Euclidean and the intrinsic distance on M. (McInnes, Healy, and Melville 2018) argue that, even if M is embedded in an Euclidean space, the induced metric might not be optimal for dimensionality reduction and one should choose a metric that also reflects the distribution of the data. In fact, for many manifold learning based dimensionality reduction techniques it is essential that the data set is sampled from the uniform distribution on the Riemannian manifold. That is, the probability of finding x in A is proportional to the Riemanian volume of A. When, however, the data are not distributed uniformly w.r.t. the Riemannian metric g, we can conformally change g to make the distribution uniform, i.e., replace g by λg, for a positive smooth function λ on M, c.f. Appendix D.

3 IsUMap

IsUMap is a special case of the more widely applicable metric realization of weighted simplicial complexes above. We start with a metric dataset $(X = \{x_1, ..., x_n\}, d)$, assumed to be sampled from a (not necessarily uniform) distribution on a Riemannian manifold (M, g). We first construct a (triangulated) metric space (Z, d_Z) which approximates the Riemannian manifold $(M, \lambda g)$. We then aim to find points $Y = \{y_1, ..., y_n\}$ in a low dimensional Euclidean space $\mathbb{R}^d$ with pairwise distances sufficiently close to the vertices of the triangulation (Z, d_Z). In the discrete setting, for a finite dataset X, the combinatorial counterpart of this procedure consists in constructing weighted (star) graphs $(\Gamma_i)_{i \in \{1,\dots,n\}}$ from the $k-$neighborhood graph of (X, d). The weight function on Γ_i is equal to the radial distances w.r.t. d scaled according to the density around x_i. We thus, akin to (McInnes, Healy, and Melville 2018), define each Γ_i as the star graph with x_i as the center and its k nearest neighbors x_{i_j}, $j = 1, \dots k$ as the outer vertices and the corresponding weight W_{ij} on the edge between x_i and x_{i_j} as $W_{ij} := \frac{d(x_i, x_j)}{\sigma_i}$ where $\sigma_i = d(x_i, x_{i_k})$. One may also subtract $d(x_i, x_{i_1})$ from all distances before normalization, to mitigate the curse of dimensionality. The graphs Γ_i corresponding to VR-filtrations of metric spaces (X, d_i), $i = 1, \dots n$ obtained by local distortion of d via

$$d_i(x_i, x_{i_j}) = d_i(x_{i_j}, x_i) = \frac{d(x_i, x_{i_j})}{\sigma_i}, \quad \text{for } j = 1, \dots k, \tag{3}$$

$d_i(x,x) = 0 \; \forall x$, and all other distances are set to ∞. These local graphs are merged and glued by the operations described in Section 2.3 to obtain a global weighted graph (Γ, W) on the vertex set X. The specific structure of the metrics (3) facilitates the merging. Each edge admits at most two weights (when both end points are each other's neighbors). The final distance d_Z corresponding to the metric realization, which is the geodesic (or graph) distance on Γ (being computed by Dijkstra's algorithm). This bottleneck causes the complexity $\mathcal{O}(N^2 \log N)$, but parallelization is possible. After merging and applying the metric realization, we

can choose a loss function $\mathcal{L}\left([d_{\mathbb{R}^k}(y_i, y_j), d_Z(x_i, x_j)]_{i,j}\right)$ of distances to obtain the set $Y = \{y_1, \ldots, y_n\}$ that minimizes $\mathcal{L}$. We use the loss function of metric MDS (Abdi 2007), but that of classical MDS (Torgerson 1952; Lee, Verleysen et al. 2007) works as well. Below we present an explanation of the steps of the pseudo-algorithm for IsUMap presented in algorithm 1, c.f. Appendices for computing infrastructure.

The IsUMap scheme begins with input data represented as a distance matrix D, where D_{ij} corresponds to the the distance between x_i and x_j in X according to the given metric d. After selecting the parameters of the algorithm, i.e. k (for construction of $k-$neighborhood graph), the operator ϕ (to map between probabilistic and metric weights), and the T-conorm T (used to symmetrize the weight matrix), and identify neighbors for each point based on the $k-$nearest neighbor criterion, it proceeds as follows

1- Create an $n \times n$ matrix W, where the entries are defined $W_{ij} := \phi(d_i(x_i, x_j))$, using the map ϕ and the local metrics d_i in (3). This yields a non-symmetric weight-adjacency matrix for the k-neighborhood graph of X.

2- Convert W into a symmetric matrix $\tilde{W}$ by replacing both W_{ij} and W_{ji} with $T(W_{ij}, W_{ji})$, where T is the pre-set (T-conorm) operator. This yields non-zero value if either W_{ij} or W_{ji} is non-zero.

3- Convert the (symmetrized matrix) $\tilde{W}$ to a metric-weighted adjacency matrix $\tilde{D}$ by applying ϕ^{-1} to non-zero entries. For entries corresponding 0 values in $\tilde{W}$, replace them with the maximum value of non-zero value (or use a computationally efficient alternative implemented in the code).

4- Create new distance matrix D_{new} (whose entries represent distances between pairs of points) by applying Dijkstra's algorithm to $\tilde{D}$.

5- Input the new metric distance matrix D_{new} into an MDS algorithm, either metric or classical, to obtain the final embedding in a low-dimensional Euclidean space.

4 Experiments

The following experiments illustrate how IsUMap does not only capture topological features, and achieves density uniformization but also preserves intrinsic geometry.

Low-Dimensional Geometries. We first test IsUMap on toy examples. In Fig. 1, we see samples of size 3000 from the Swiss Roll with a hole and the Möbius strip, and in Fig. 2, a sample drawn from a 3-dimensional Mammoth with 20000 points. While the distributions of the Swiss Roll and Mammoth are uniform, the Möbius strip sample is intentionally non-uniform to highlight IsUMap's ability to handle complex datasets that combine non-trivial topology with distribution irregularities. For low-dimensional datasets, we opted to exclude the subtraction of the distance to the nearest neighbor (ρ_i) in the expression of local metrics (3), as the curse of dimensionality does not affect distances in these cases.

Algorithm 1 IsUMap

Input: Samples $x_1, ..., x_n$ and/or distances $d(x_i, x_j)$
Parameter: Metric-to-weight-function ϕ, k-neighborhood, T-conorm T, normalization (boolean)
Output: Low dimensional representations $y_1, ..., y_n$.

initialize distance matrix D
knn,knn_indices = k_nearest_neighbors(X,d)
D[not knn_indices] = ∞
if normalization = True **then**
    D = D/D[:,knn_indices[k]]. {normalize by distance to k-th neighbor for uniformization}
end if
W = $\phi(D)$
$\tilde{W} = T(W, W^T)$ {combine different edge weights via t-conorm}
$\tilde{D} = \phi^{-1}(\tilde{W})$
D_{new} = Dijkstra($\tilde{D}$)
$y_1, ..., y_n$ = MDS(D_{new})
return $y_1, ..., y_n$

Incorporating UMAP's emphasis on local relationships and Isomap's geodesic distance computation, IsUMap effectively unfolds the Swiss Roll with hole dataset, but preserves the intrinsic geometry of both local and global structures as well as the topological feature (the hole) and maintains cluster integrity. In comparison, Isomap relies on geodesic distances, and the discontinuous visualization by UMAP highlights its emphasis on preserving local structures at the expense of overlooking some global patterns. Visualization of the sample with non-uniform distribution on Mobius strip generated by nonuniform angle around strip using IsUMap in Fig. 1, shows its attempt to uniformize this distribution, in contrast to Isomap, which retains the original density. Additionally, IsUMap preserves both the shape and topological features similar to Isomap's result, while providing a more accurate representation than UMAP. The Mammoth dataset tests IsUMap on a low dimensional dataset with more complicated geometry. The outcome in Fig. 2 shows how IsUMap well preserves the clusters, geodesic distances and the shapes of body and bones and spreads out the different body protrusions, significantly better than UMAP and Isomap.

Non-uniform Data. We shall now analyze how IsUMap can deal with non-uniform distributions. As discussed in Section 2, we apply a conformal transformation to the metric with the pointwise scaling factor σ. Here, we consider such an example on a simple geometric object, namely a hemisphere, allowing us to focus on distribution irregularities without the added complexity of topological non-triviality, as in the Möbius strip. Since the dimension is reduced only by 1, we don't need to subtract $d(x_i, x_{i_1})$. We compare the resulting embedding with that of UMAP, which also uses a local scaling of the metric, but combines the local metrics differently, and with the Isomap method, which does not scale. See Fig. 3 for the dataset and the mappings to $\mathbb{R}^2$

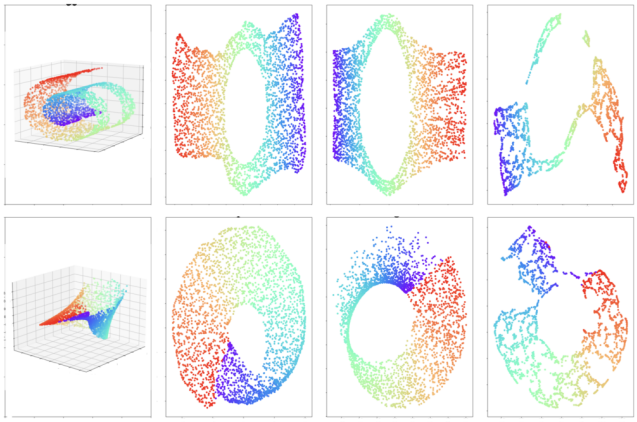

Figure 1: Visualization of the Swiss Roll with hole and the Möbius strip in $\mathbb{R}^2$ with columns from left to right, initial data, IsUMap, Isomap, UMAP

by these methods. Our dataset has a denser region around the pole and sparser regions towards the equator. Also, as the hemisphere has a boundary, we expect boundary effects. As the Fig. 3 shows, IsUMap creates a projection that looks like a disk with uniformly distributed data points in the interior. While UMAP seems to follow a similar strategy, it introduces discontinuities. Isomap represents the data with a more denser distribution in the center of the disk, like a projection/top view of the data.

4.1 Computational Biology Experiments

We next apply IsUMap to some high dimensional biological datasets, with a-priori unknown underlying manifold structure. We analyze the datasets of trefoil-knotted protein chains building on the work of (Benjamin et al. 2023), when one of the main tasks after dimensionality reduction is clustering, and single-cell RNA, for which trajectory inference and pseudo-time tasks using RNA velocity (La Manno et al. 2018) will be performed after dimensionality reduction.

Trefoil-Knotted Protein Chains. In the first example, we analyze trefoil-knotted protein chains by employing Wasserstein and L_1 distances on their persistence diagrams and persistence landscapes (Bubenik et al. 2015), resp. The mathematical pipeline of (Benjamin et al. 2023) analyzes the geometric features and topological dissimilarity of protein entanglement with persistent homology and Isomap, and so, we can compare that with UMAP and IsUMap.
The dataset consists of proteins with backbones forming open-ended positive trefoil knots, labeled once by structural homology classes based on sequence similarity and next by their knot depth (shallow, deep, or neither). Although one may analyze the dataset (with either of labelings) using Wasserstein and L_1 distance on the persistence diagram and persistence landscapes, resp., as suggested in (Benjamin et al. 2023), it seems both representations are equivalent from the geometric perspective, as can be seen in Appendix E.2,I. Here, we present only the results for one initial representation for each labeling method.

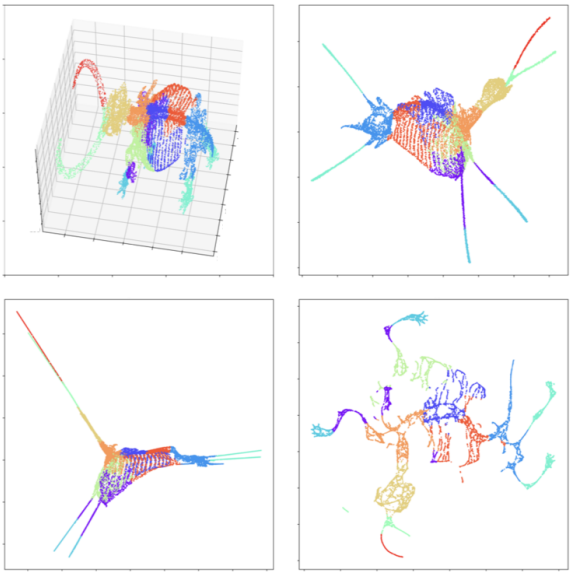

Figure 2: Visualization of mammoth dataset (top-left) in $\mathbb{R}^2$ by IsUMap (top-right), Isomap (bottom-left) and UMAP (bottom-right).

We apply IsUMap and UMAP to labeling by structural homology classes, using the L_1 distance on its persistence landscape, and to labeling by depth category, with the Wasserstein distance on its persistence diagram, recorded in Fig. 4. UMAP effectively clusters the dataset for both distances, but does not distinctly reveal the underlying structures of the protein space. In contrast, IsUMap not only shows the clusters, but also brings out the tripod shape characteristic of trefoil-knotted proteins and achieves a uniform distribution, which prevents data points from overlapping, a common issue in UMAP and, to some extent, in Isomap representations used in (Benjamin et al. 2023).

Trajectory Inference from Single-Cell RNA Data. We next consider trajectory inference and pseudo-time tasks using RNA velocity (La Manno et al. 2018), a method approximating the time derivative of gene expression. This approach is applied to infer the maturation trajectories of neural progenitor cells into various neural cell types, emphasizing continuity in underlying structures. Leveraging the human forebrain dataset from (La Manno et al. 2018), which explores transcriptional dynamics during brain development, we utilize IsUMap and UMAP to project the data into $\mathbb{R}^2$, and then obtain RNA velocity from this data, using (La Manno et al. 2018).
IsUMap achieves a more uniform dataset distribution and shows connected geodesic paths, thereby preserving the expected 'true' structure of the underlying developmental dynamics. Fig. 5 compares UMAP and IsUMap results in trajectory inference, similar to (Chari and Pachter 2023). UMAP has problems in representing continuous relationships and in establishing a stage for the trajectory inference as a downstream task (which has been attributed primarily due to its over-reliance on a negative sampling step (Dam-

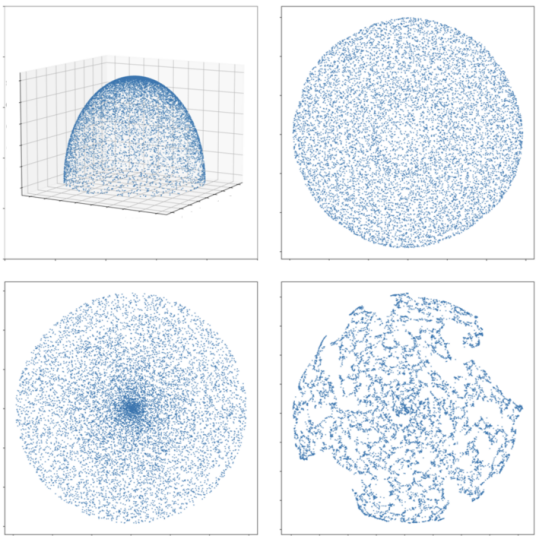

Figure 3: Visualization of a sample of $N = 10000$ points on a hemisphere with non-uniform distribution (top-left) in $\mathbb{R}^2$ with $k = 30$ by IsUMap (top-right), Isomap (bottom-left) and UMAP (bottom-right)

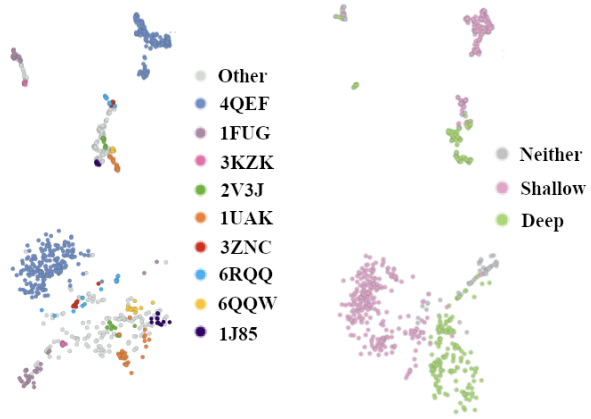

Figure 4: Representations of Trefoil-knotted protein chains, labeled by sequence homology classes with Wasserstein distance (left) and depth category with L_1 distance (right) **Top**: UMAP embedding, **Bottom**: IsUMap embedding, both with k=15 in neighborhood graph

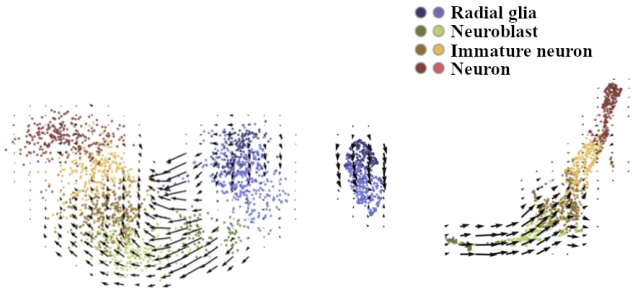

Figure 5: Trajectory Inference in IsUMap (left) and UMAP (right).

rich and Hamprecht 2021a)). In contrast, applying RNA velocity to representations from IsUMap leads to improved continuity in underlying structures and a more accurate trajectory inference.

4.2 Topological Inference in Gridcell Dataset

We next analyze neural recordings of grid cells with the aim of identifying the topology of population codes. We build on (Gardner et al. 2022), but replace PCA with either IsUMap or UMAP for dimensionality reduction. Fig. 6 presents the results of dimensionality reduction from 93 to 3 using IsUMap and UMAP (top row), followed by how each method can detect the toroidal topology of this dataset (bottom row). As shown in both visualizations and persistence diagrams, IsUMap captures this topology more accurately, whereas UMAP primarily emphasizes cluster separation, cf. Appendix E.2,III. That is, IsUMap, in contrast to UMAP, shows the correct single component, while the cycles are captured in the (appropriately reduced) figure. For this dataset, prior studies have already established the topological profile (cf. (Gardner et al. 2022)), and our primary focus was to evaluate the efficiency of IsUMap compared to UMAP in preserving this profile. However, in general, TDA methods, such as persistent homology, can serve as reliable tools for assessing the quality of dimensionality reduction outcomes, as discussed in (Rieck and Leitte 2015).

4.3 IsUMap and Clustering/Classification

Finally, we compare the clustering abilities of IsUMap with the other methods. While not the primary objective of methods like UMAP, clustering is frequently relevant in practise. We use two standard high dimensional benchmarks, MNIST and the Wisconsin breast cancer datasets. Fig. 7 shows the result of IsUMap, Isomap and UMAP on the MNIST dataset, colored by ground truth. IsUMap separates clusters more distinctly than Isomap, which tends to mix clusters, but not as effectively as UMAP. To systematically explore this, we applied the Pair Sets Index (PSI) introduced in (Rezaei and Fränti 2016), to evaluate the effect of these three dimensionality reduction methods in clustering of the MNIST dataset. A higher value in $[0, 1]$ indicates a better labeling with respect to ground truth annotations. For all cases, we used $k-$means as the clustering method with $k = 10$, and PSI assessed how the resulting clusters aligned with the initial labeling by 10 digits of the data. Fig. 8 shows the PSI graph evaluated across multiple dimensions, linearly interpolated in between. We want to determine whether clustering performance improves as the dimension reduces and when further dimension reduction negatively impacts the task. UMAP performs best in clustering, followed by IsUMap; Isomap is the worst. All three methods achieve the best clustering in dimension 10, with a sharp decline below 10. Similarly, in Fig. 9, we compare the three methods on the Wisconsin breast cancer dataset in $\mathbb{R}^2$. Unlike Isomap, IsUMap separates two clusters (benign and malignant) well. Moreover, since IsUMap aims to preserve the geodesic distances between data points based on the local distances (3), its un-

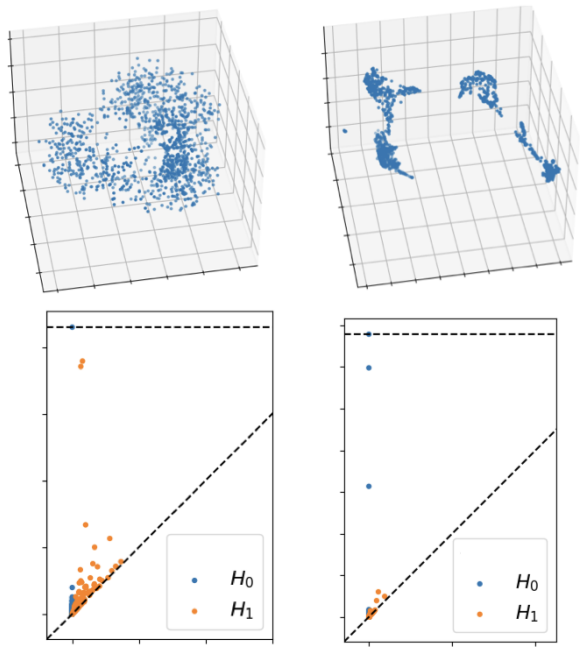

Figure 6: Visualization and persistence diagram of grid cell dataset after dimensionality reduction by IsUMap (left) and UMAP(right).

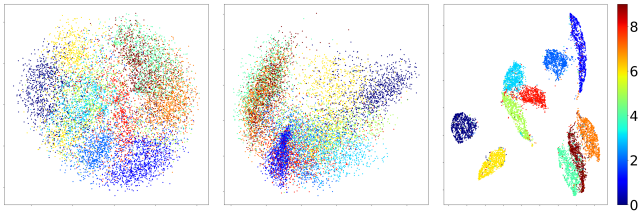

Figure 7: Clusters in MNIST (data size $= 10000$) after DR by (left to right) IsUMap, Isomap and UMAP.

derlying shape may provide a clearer representation of the structure of the dataset than UMAP.

It seems that the clustering capabilities of UMAP mainly stem from how the embedding is performed, in particular from the negative undersampling, cf. (Damrich and Hamprecht 2021b). We, therefore, conjecture that a different embedding method than pure MDS might improve the clustering of IsUMap as well.

To further investigate the performance of IsUMap, we turn to supervised learning, namely classification. We trained a linear classifier on the representations obtained from IsUMap, UMAP, Isomap on a subset of the CIFAR-10 dataset (Krizhevsky, Hinton et al. 2009), varying the embedding dimension between 2 and 400, and report the test accuracy of each method (see appendix for training hyperparameters) in Fig. 10. IsUMap and Isomap consistently outperform UMAP here. Although Isomap performs slightly better in lower dimensions, the best performance is achieved in higher dimensions, and here IsUMap outperforms both Isomap and UMAP. It seems that IsUMap retains more useful information about cluster labels than the other methods.

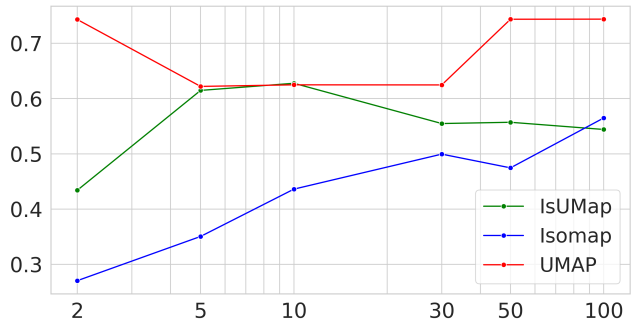

Figure 8: PSI across various dimensions for MNIST after dimesionality reduction by different methods.

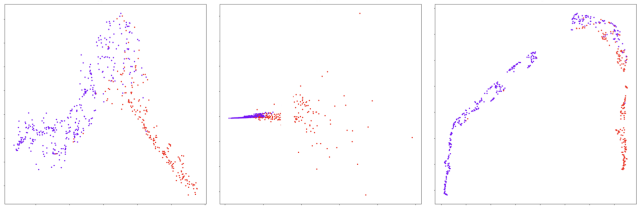

Figure 9: Clusters in the Wisconsin dataset (data size 570, dim$=$ 32, $k = 20$), after DR by (left to right) IsUMap, Isomap and UMAP.

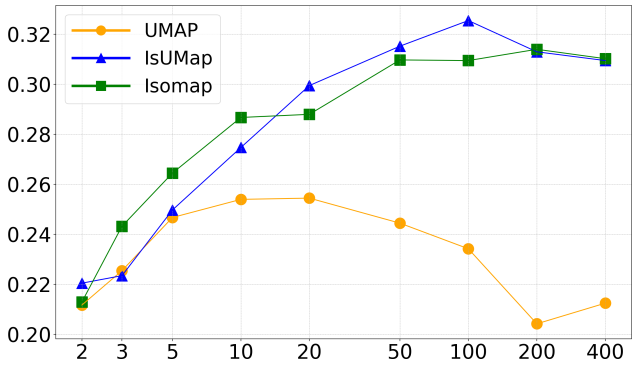

Figure 10: Test accuracy of a linear classifier across various dimensions for CIFAR-10 (initial dimension $= 1024$) after dimensionality reductions by different methods.

4.4 Conclusion

On the basis of a general theory of metric realization of weighted simplicial complexes via fuzzy constructions, we derived IsUMap, a method which leverages this theory to construct low dimensional representations. IsUMap can, more faithfully than related methods, represent a geometry where the data is not uniformly distributed. We presented applications in computational biology. As for all existing manifold learning methods, the quality of the low dimensional representations obtained by IsUMap depends on the intended downstream task, suggesting corresponding adjustments of the final step.

Acknowledgements

We thank Eckehard Olbrich, Armin Pournaki, Abel Jansma, and Sayan Mukherjee for inspiring discussions.

References

Abdi, H. 2007. Metric multidimensional scaling (MDS): analyzing distance matrices. *Encyclopedia of measurement and statistics*, 1–13.

Amid, E.; and Warmuth, M. K. 2019. TriMap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204*.

Barth, L. S.; Fahimi, F. H.; Joharinad, P.; Jost, J.; Keck, J.; and Mikhail, T. J. 2024. Fuzzy simplicial sets and their application to geometric data analysis. *https://arxiv.org/abs/2406.11154*.

Belkin, M.; and Niyogi, P. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15: 1373–1396.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828.

Benjamin, K.; Mukta, L.; Moryoussef, G.; Uren, C.; Harrington, H. A.; Tillmann, U.; and Barbensi, A. 2023. Homology of homologous knotted proteins. *Journal of the Royal Society Interface*, 20(201): 20220727.

Botteghi, N.; Poel, M.; and Brune, C. 2022. Unsupervised representation learning in deep reinforcement learning: A review. *arXiv preprint arXiv:2208.14226*.

Bubenik, P.; et al. 2015. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1): 77–102.

Burago, D.; Burago, Y.; and Ivanov, S. 2001. *A course in metric geometry*. AMS.

Carlsson, G. 2009. Topology and Data. *Bulletin of the American Mathematical Society*, 46: 255–308.

Chari, T.; and Pachter, L. 2023. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8): e1011288.

Damrich, S.; and Hamprecht, F. A. 2021a. On UMAP's true loss function. *Advances in Neural Information Processing Systems*, 34: 5798–5809.

Damrich, S.; and Hamprecht, F. A. 2021b. On UMAP's true loss function. *Advances in Neural Information Processing Systems*, 34: 5798–5809.

Gardner, R. J.; Hermansen, E.; Pachitariu, M.; Burak, Y.; Baas, N. A.; Dunn, B. A.; Moser, M.-B.; and Moser, E. I. 2022. Toroidal topology of population activity in grid cells. *Nature*, 602(7895): 123–128.

Hausmann, J. C. 1995. On the Vietoris-Rips complexes and a cohomology theory for metric spaces. *Annals of Mathematics Studies*, 138: 175–188.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

La Manno, G.; Soldatov, R.; Zeisel, A.; Braun, E.; Hochgerner, H.; Petukhov, V.; Lidschreiber, K.; Kastriti, M. E.; Lönnerberg, P.; Furlan, A.; et al. 2018. RNA velocity of single cells. *Nature*, 560(7719): 494–498.

Latschev, J. 2001. Vietoris-Rips complexes of metric spaces near a closed Riemannian manifold. *Archiv der Mathematik*, 77(6): 522–528.

Lee, J. A.; Verleysen, M.; et al. 2007. *Nonlinear dimensionality reduction*, volume 1. Springer.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Https://arxiv.org/abs/1802.03426.

Rezaei, M.; and Fränti, P. 2016. Set matching measures for external cluster validity. *IEEE transactions on knowledge and data engineering*, 28(8): 2173–2186.

Rieck, B.; and Leitte, H. 2015. Persistent homology for the evaluation of dimensionality reduction schemes. In *Computer Graphics Forum*, volume 34, 431–440. Wiley Online Library.

Spivak, D. I. 2009. Metric realization of fuzzy simplicial sets. *N.A.* Https://math.mit.edu/ dspivak/files/metric_realization.pdf.

Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500): 2319–2323.

Torgerson, W. S. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4): 401–419.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Zomorodian, A.; and Carlsson, G. 2004. Computing persistent homology. *ACM*, 347–356.