TopoMap: A 0-dimensional Homology Preserving Projection of High-Dimensional Data

Harish Doraiswamy, Julien Tierny, Paulo J S Silva, Gustavo Nonato, Claudio Silva

TopoMap: A 0-dimensional Homology Preserving Projection of High-Dimensional Data

Harish Doraiswamy, Julien Tierny, Paulo J. S. Silva, Luis Gustavo Nonato, and Claudio Silva

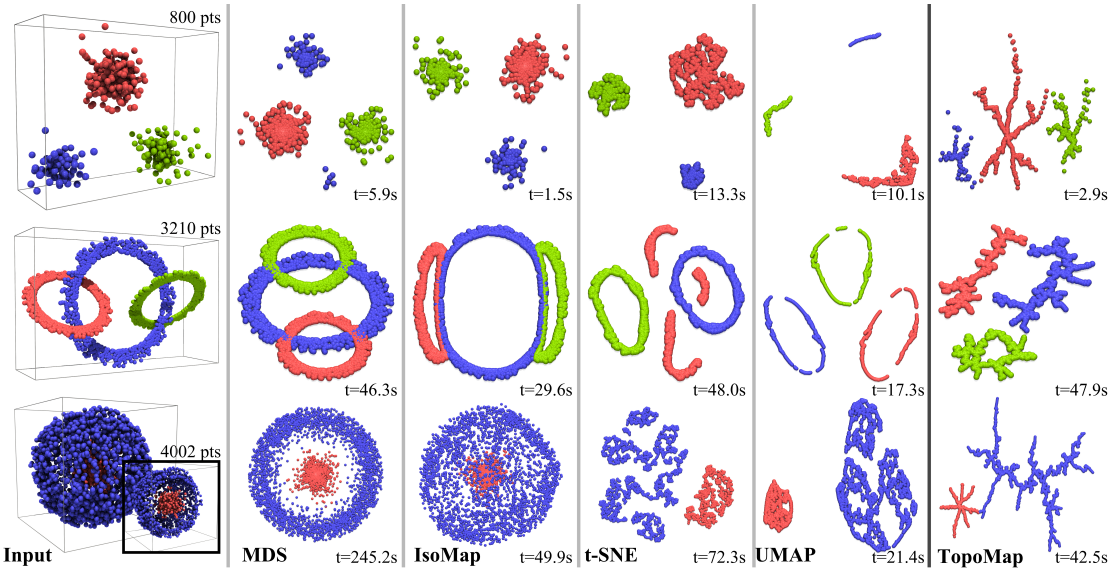

Fig. 1. The result of mapping three dimensional data to a 2D space using geometry preserving projections: Classical MDS, Isomap, tSNE, UMAP; and the proposed topology preserving TopoMap method. While geometry preserving methods tend either to split connected components or mix them up, TopoMap is guaranteed to preserve them, leveraging more reliable analysis.

Abstract— Multidimensional Projection is a fundamental tool for high-dimensional data analytics and visualization. With very few exceptions, projection techniques are designed to map data from a high-dimensional space to a visual space so as to preserve some dissimilarity (similarity) measure, such as the Euclidean distance for example. In fact, although adopting distinct mathematical formulations designed to favor different aspects of the data, most multidimensional projection methods strive to preserve dissimilarity measures that encapsulate geometric properties such as distances or the proximity relation between data objects. However, geometric relations are not the only interesting property to be preserved in a projection. For instance, the analysis of particular structures such as clusters and outliers could be more reliably performed if the mapping process gives some guarantee as to topological invariants such as connected components and loops. This paper introduces *TopoMap*, a novel projection technique which provides topological guarantees during the mapping process. In particular, the proposed method performs the mapping from a high-dimensional space to a visual space, while preserving the 0-dimensional persistence diagram of the Rips filtration of the high-dimensional data, ensuring that the filtrations generate the same connected components when applied to the original as well as projected data. The presented case studies show that the topological guarantee provided by TopoMap not only brings confidence to the visual analytic process but also can be used to assist in the assessment of other projection methods.

Index Terms—Topological data analysis, computational topology, high-dimensional data, projection.

1 Introduction

Multidimensional Scaling (MDS) accounts for the problem of embedding data in a Cartesian space while preserving intrinsic properties of the data. A particularly important task in the context of MDS is dimensionality reduction, which aims to map data from a d-dimensional to a k-dimensional Cartesian space where $k << d$. In the context of visualization, where the embedding space is 2D or 3D, MDS is typically called multidimensional projection (MDP).

Over the last decades, a multitude of MDP methods have been developed to map high-dimensional data to a visual space while preserving geometric properties such as the Euclidean distance between data objects. A main issue shared by all those methods is that the preservation of geometric properties can only be guaranteed under very particular conditions. Thus, errors and distortions are highly likely in the resulting mapping, introducing uncertainties to analytical procedures carried out from projection layouts [68]. For instance, structures observed in the point cloud resulting from a projection such as neighborhood relations might not be the ones existing in the original data, thus potentially leading inexperienced practitioners to wrong conclusions.

Although a number of alternatives have been proposed to render the analysis of projection layouts more reliable [54], few are focused on developing MDP methods with theoretical guarantees as to properties

- *H. Doraiswamy and C. Silva are with New York University; J. Tierny is with CNRS and Sorbonne Université; P. J. S. Silva is with University of Campinas; and L. G. Nonato is with University of Sao Paulo, Sao Carlos.*
- *E-mail: {harishd,csilva}@nyu.edu, julien.tierny@sorbonne-universite.fr, pjssilva@ime.unicamp.br, gnonato@icmc.usp.br*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

preserved by the mapping. Guaranteeing that a certain property is preserved exactly by the mapping makes the analytical process more reliable and meaningful, ensuring that what is seen is indeed what takes place in the high-dimensional space.

This work introduces TopoMap, a novel MDP technique that is guaranteed to preserve topological structures during the dimensionality reduction process. Specifically, TopoMap maps high-dimensional data to a visual space while preserving 0-homology (Betti-0) topological persistence as defined by a Rips filtration over the input data points. Intuitively, a Rips filtration grows a high dimensional ball around the data points, and adds an edge (or a high dimensional simplex) to the filtration when two (or more) balls intersect. In other words, the proposed method ensures that the topological filtrations over both the original as well as the projected data generate the same connected components at the same instances of the respective filtrations. The topological guarantee provided by TopoMap allows analysts to confidently explore high-dimensional data by visualizing which groups of objects are more tightly connected in the high-dimensional space. As we show in the provided case studies, visualizing persistent components via Betti-0 preserving projections enables an intuitive analytical process, making the identification of objects with similar properties an easier task. Moreover, in contrast to many distance-based (dissimilarity-based) projections, there is no uncertainty in the visual identification of groups (clusters) in the layout produced by TopoMap.

Besides enabling reliable mechanisms for data exploration, the proposed methodology can be used to assess and better understand distance-based projections. Since TopoMap is guaranteed to preserve the connected components of a particular neighborhood graph structure, one can rely on it to analyze how those connected components are mapped by other projection methods. As a result, one can further understand how distance-based MDP methods split or merge components, thus revealing regions with distortion.

In summary, the main contributions of the work are:

- A theoretical framework to support the design of a dimensionality reduction technique called TopoMap, which is guaranteed to preserve the Betti-0 topological persistence defined by the Rips filtration over the data.
- An optimization procedure that ensures the correct mapping of the connected components resulting from the filtration process.
- An exhaustive evaluation using both labeled data as well as case studies over unlabeled data showing the potential of TopoMap to support the analysis of high-dimensional data as well as distance-based projection techniques.

To the best of our knowledge, TopoMap is the first dimensionality reduction technique to provide guarantees as to the preservation of the topological properties of the Rips filtration of the data under analysis.

2 Related Work

In order to better contextualize the proposed methodology, we organize the related work in two main parts, topological data analysis (which describes related work in topological data representations) and topology-based multidimensional projection (which describes how these topological data representations can drive projection methods).

2.1 Topological data analysis

Topology-based methods [24] have been very popular in the last two decades to support advanced data analysis and visualization tasks [37]. By providing a concise, structural representation of the data, these techniques greatly help in the visualization and analysis of the data. They have been applied successfully to a variety of domains, such as astrophysics [69, 73], biological imaging [2, 9, 14], chemistry [8, 32, 59], fluid dynamics [41], material sciences [34, 35, 72], or turbulent combustion [11, 33, 44].

The Rips filtration [5, 24] is often used to analyze the topology of high dimensional point clouds, and is motivated by the work by Chazal and Oudot [16] who showed that Rips filtrations can provably capture the homology of the manifold sampled by the point cloud.

Among the popular representations in topological data analysis, the *Reeb graph* [63] is obtained by contracting to a single point each of the connected components of level sets of an input scalar field, resulting in a characteristic skeleton-like representation of the input data. For discrete point cloud data, the *Mapper* [71] is an approximation of the Reeb graph of some user-defined function (often called *lense* function) defined over a nearest neighbor graph of the input point cloud. Another popular abstraction is the Morse-Smale complex [21], which is a cellular decomposition of the domain of an input scalar field, such that all the points of a given cell admit the same gradient integration extremities. For discrete point cloud data, the Morse-Smale complex has been used over the k-nearest neighbor graph of the input point cloud for clustering purposes [15]. As discussed next, all of these representations (Mapper, Reeb graph, Morse-Smale complex) have also been used as a driving data representation for dimensionality reduction.

Given the increasing popularity in using topology-based techniques for data analysis, it is not surprising that there are several open source tools and libraries available [1, 6, 12, 27, 49, 52, 53, 75].

2.2 Topology-based Multidimensional Projection

Multidimensional projection has long been a fundamental analytical tool, mainly in the context of visualization [42, 54]. In fact, the visualization community has not only proposed a number of MDP methods tailored to visual analytic tasks [39], but has also developed methodologies to facilitate the analysis of MDP distortions [3, 50] and to enrich MDP layouts so as to uncover information hidden in the projections [31, 40]. The extensive literature about MDS/MDP techniques has been organized over several books [10, 47] and surveys [19, 54, 77]. In order to emphasize our contribution, we focus only on techniques that explicitly rely on topological concepts to perform and assess multidimensional projections, disregarding distance preserving methods such as the classical MDS [47] and neighborhood preserving techniques such as LLE [66], t-SNE [76], and Lamp [39]. We refer interested readers to the above books and surveys for a broader discussion about MDS/MDP methods.

Isomap [74] is prossibly one of the first MDS techniques to resort to topological mechanisms to accomplish dimensionality reduction. Isomap aims to capture the topological (manifold) structure of the data through a graph representation from which geodesic distances are computed. A number of variants of Isomap have been proposed, including Landmark (L-Isomap) [70], out-of-sample [7] and spatio-temporal extensions [38]. An interesting variant of Isomap is the method proposed by Lee and Verleysen [46], which tears a graph representation of the data so as to preserve essential (non-contractable) loops, thus enabling loop preserving manifold unfoldings. The recent work by Yan et al. [79] is another particularly interesting variant of Isomap (precisely, a variant of L-Isomap). Similar to previous work on skeletonization [43], this approach identifies *cycles* in the original data, but additionally aims at preserving these cycles when projecting the data to 2D. Specifically, it focuses on the Mapper (cf. Sect. 2.1) of a function defined on the KNN-graph structure of the data to select landmark points. The underlying idea is that the topology-based landmark selection captures the structure of the 1-dimensional homology groups of the data, which hopefully are preserved during the dimensionality reduction phase accomplished via regular L-Isomap. However, their approach does not take into account 0-homology groups which are therefore not preserved. In contrast, the TopoMap method proposed in this work provides theoretical guarantees as to 0-dimensional homology group preservation, thus ensuring that the connected components visualized in the projection layout are the same as in the original high-dimensional data, according to its Rips filtration. Similar to Yan et al. [79], Gerber et al. [29, 30] introduced projection methods driven by topological data representations. In particular, it differs from our work in the sense that the introduced projections are driven by the network of cells of maximum dimension (called crystals) of the Morse-Smale complex (Sect. 2.1). They do not aim at specifically preserving the persistence diagram of the Rips complex as studied in this paper and therefore encode a different information, specifically tailored for regression tasks.

In scientific visualization, Weber et al. [78] introduced a terrain

metaphor to provide an intuitive visualization of the topological features present in a volume scalar field. Due to occlusion, these features can be challenging to visualize when represented in their original 3D space. This work addresses this issue by constructing a 2D terrain whose elevation is carefully designed, such that the contour tree of the elevation map matches the contour tree of the original data in 3D. The resulting elevation can also be displayed as a planar heat map and the original data points can in principle be projected to this planar layout, by inserting each 3D point in the 2D region corresponding to its arc in the contour tree. This method can be interpreted as topology preserving, as the contour tree of the 2D heatmap is guaranteed by construction to be equal to the contour tree of the original data in 3D. Note however, that the algorithm for constructing the terrain solely focuses on the contour tree and ignores the metric information coming from the original data. In particular, it places the root of the branch decomposition of the contour tree at the center of the layout and then arranges the children branches along a spiral trajectory [78]. This can have the effect of projecting in a small 2D neighborhood topological features which were originally arbitrarily far apart in 3D. Harvey and Wang [36] proposed algorithms to generate an ensemble of terrains each having the same contour tree as the input data. However, the shortcomings described above apply to these terrains as well.

In a series of papers [55–58], Oesterling et al. extended this approach to the case of high-dimensional point clouds. This line of work is probably the most related to our approach. When extending the terrain metaphor to such data, the first difficulty is to derive a simplicial representation of the point cloud. In their work, Oesterling et al. suggest to use a specific adjacency graph called the Gabriel graph [28]. The second challenge consist in deriving a scalar field on this graph which faithfully describes the data. The authors opt for a kernel density estimation of the point cloud (with a Gaussian kernel). From this point, the terrain metaphor [78] can be applied and the authors introduce various improvements [56, 58] based on contour profiles for instance [57].

In our work, by considering the Rips filtration, we focus our analysis on *distances*, while Oesterling et al. focus on *densities*. In that regard, these two approaches are complementary, just like distance-based and density-based clustering methods are complementary. More importantly, the two approaches differ in the way the layout of the data in 2D is computed. As discussed above, the terrain metaphor [78] provides a constructive approach for computing the output 2D layout which discards the metric information of the original space, as acknowledged by the authors [55]. Data points which are arbitrarily far in the original space can be projected arbitrarily close, and reciprocally. In contrast, our layout strategy enforces the preservation of the persistent homology of the Rips filtration. This enables to better take into account the metric properties of the data, and to some extent be more faithful to its original geometry. This tends to preserve the spatial relations between clusters (which are not taken into account in terrain metaphors). For instance, in Fig. 1, the central clusters in the data (top row: red, middle row: blue) are indeed projected in between the other clusters with our method (top and middle row, right column).

A subtle, yet important, distinction between our work and terrain metaphors [55–58] is that our approach preserves topological features *strictly* when projecting the data to 2D. In particular, the 0-dimensional persistence diagram of the Rips filtration of the projected data is strictly equal to that of the high dimensional data, by construction. In contrast, terrain metaphors for high dimensional data [55–58] provide topology-preserving *terrains*, but not necessarily topology-preserving *projections*, as each data point is placed *"at a random position along its* (density) *contour"* [55]. Finally, note that to our knowledge, no public implementation of the terrain metaphors is available.

The recently introduced UMAP approach [51] is based on topological notions, namely category theory while our approach focuses on Persistent Homology [24]. As reported by its authors, UMAP provides visual results which are highly similar to t-SNE. For this reason, it is often regarded as a faster, more modern and more scalable alternative to t-SNE, which still provides visually similar outputs.

Topological tools have also been the basis of methods designed to evaluate the quality of dimensionality reduction techniques. A good example is the work by Rieck and Leitte [65], which assesses the quality of a projection technique from the 2nd Wasserstein distance between persistence diagrams computed from the original and projected data. In a follow up work, Rieck and Leitte [64] proposed the use of persistent homology to compare quality measures for dimensionality reduction, making possible to analyze the agreement of multiple quality measures, thus identifying regions where different quality measures disagree the most. Persistent homology has also been employed by Paul and Chalup [60] as a mechanism to validate dimensionality reduction methods when applied to particular benchmark data. As we shall prove later, TopoMap is guaranteed to preserve connected components under filtration, and is therefore exact (no error) when comparing the 0-dimensional homology persistence diagrams generated by a filtration in the visual and original spaces respectively.

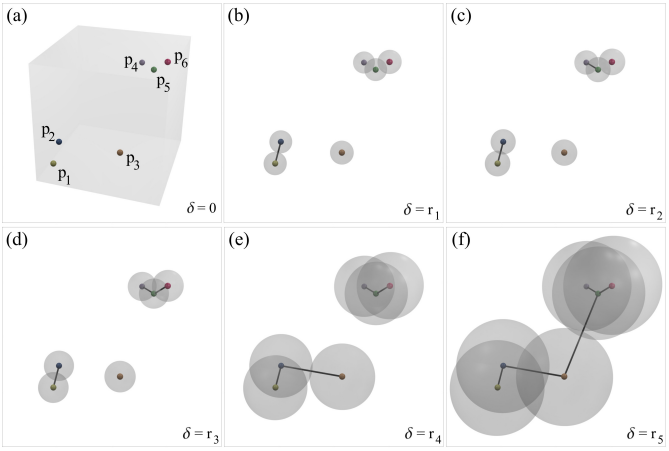

Fig. 2. The ball growth model used to analyze the topological properties of point data sets. **(a)** Input data. **(b)–(f)** Different stages of the filtration with increasing diameter δ. These stages correspond to the instant in the filtration when two components (0-cycles) merge into one. The edge from the Rips filtration responsible for this merge is also shown. Note that this collection of edges correspond to the minimum spanning tree of the input points.

3 Topology Preserving Projection

Given a data set that is a collection of high-dimensional points in $\mathbb{R}^d$, a common topology-based approach to analyze this data is to study the evolution of cycles in the simplicial complexes resulting from a Euclidean distance based Rips filtration over these points. Our goal is to project the data onto $\mathbb{R}^2$ such that above evolution for a subset of the cycles is preserved in the projected space as well.

In this section, we first introduce the necessary notations and formalize the problem that is of interest in this work. We refer the reader to Edelsbrunner and Harer [24] for a comprehensive discussion on these topics. Next, we describe a high level approach for solving the problem, and discuss different choices that can be made in the implementation of the high level solution.

3.1 Problem Formulation

VietorisRips complex. Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in $\mathbb{R}^d$. Given a distance threshold δ, the VietorisRips complex [24], (or Rips complex), is defined as the set of all k-simplexes $K \subseteq P$, $|K| = k+1, k \geq 0$, such that $d(p_i, p_j) \leq \delta$, $\forall p_i, p_j \in K$. Here, $d(\cdot,\cdot)$ is the Euclidean distance. Intuitively, the Rips complex for a distance threshold δ captures the shape of the data when each point p_i is replaced with a d-dimensional ball of diameter δ centered around it. For example, consider the 6 points in $\mathbb{R}^3$ shown in Fig. 2(a). Fig. 2(b)–2(f) illustrates this shape for 5 different values of the distance threshold δ.
Rips Filtration. Consider a model where the distance threshold δ is increased from 0 to ∞. That is, the d-dimensional balls are gradually grown in size. A Rips filtration captures this growth model.

Consider an ordered set of simplexes $\mathbb{K}_P = \{K_0 = \emptyset, K_1, K_2, \ldots, K_m\}$. Let δ_i, $i \in [0, m]$, be the smallest distance threshold such that simplex

K_i is part of the Rips complex defined for δ_i. Then, the above ordered set is a Rips filtration if $\forall i, j, i < j$:

1. $\exists l \leq i$ s.t. $K_i \bigcap K_j = K_l$; and
2. $\delta_i \leq \delta_j$.

Topological Persistence and Persistence Diagram. Consider the growth as defined by the Rips filtration, wherein the simplexes from the filtration are added one at a time. That is, the i^{th} iteration in this growth will consist of the subset $S_i = \{K_0, K_1, \ldots, K_{i-1}\}$. The addition of each new simplex can change the topology of the underlying data, where the topology is captured by the set of cycles in the simplicial complex defined by S_i. More specifically, a new k-cycle, $k \geq 0$, can either be created or an existing k-cycle can be destroyed [25]. Informally, a 0-cycle corresponds to a connected component, a 1-cycle to a loop, 2-cycle to a void, and so on. Given one such k-cycle, let δ_c be the threshold at which this cycle is created, and δ_d the threshold at which it is destroyed. Then the *topological persistence* [25] of this k-cycle is defined as $\delta_d - \delta_c$, and intuitively captures the lifetime of this cycle in the given filtration. Note that a cycle that is not destroyed has a persistence equal to infinity.

The *persistence diagram* [17] plots all the cycles created during the filtration as a scatter plot, where the coordinates of the point corresponding to a cycle is its creation and destruction thresholds (i.e., the x- and y-axes of this plot corresponds to the creation and destruction thresholds).

Problem Definition. Let PD_P^k denote the persistence diagram restricted to k-cycles computed using the Rips filtration over the point set P. Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^d$, our goal is to compute a corresponding set of points $P' = \{p'_1, p'_2, \ldots, p'_n\}$ in $\mathbb{R}^2$ such that $PD_P^0 = PD_{P'}^0$, where there is a one to one correspondence between the connected components or 0-cycles (i.e., a point p_i belongs to a 0-cycle w.r.t P if, and only if, the point p'_i will belong to the corresponding 0-cycle w.r.t. P').

In other words, the Rips filtration over the projected points P' not only has the exact same connected components during each iteration of the growth, but even the iterations at which they are created and destroyed are the same when compared to the Rips filtration over the high-dimensional points P.

3.2 Approach

Since we are interested only in the evolution of the set of connected components, it is sufficient to consider only the 0- and 1-simplexes (vertices and edges respectively) of the filtration. Consider the set of edges in the above filtration. Only a subset of these edges result in a change in topology, or in other words, merge two disconnected components into a single component. The following lemma bounds the number of such *topology changing edges* in the filtration.

Lemma 1. *Given a Rips filtration defined over a set of n points, there is exactly $n-1$ topology changing edges that result in reducing the number of 0-cycles.*

Proof. Consider an input with n points. At the beginning of the filtration, say at an infinitesimally small threshold $\varepsilon > 0$, there are a total of n components each corresponding to an input point. The addition of each topology changing edge reduces the count of connected components by one. Thus, there exists exactly $n-1$ such edges until there exists just a single connected component. □

The 5 edges in Figures 2(b)–2(f) correspond to the topology changing edges in the Rips filtration over the 6 points in Fig. 2(a).

Let $\mathbb{K}_P^0 = \{\emptyset, p_1, p_2, \ldots, p_n, e_1, e_2, \ldots, e_{n-1}\} \subset \mathbb{K}_P$ be the subset of a filtration, where p_i, $1 \leq i \leq n$, are the set of input points and e_i, $1 \leq i < n$, corresponds to topology changing edges (in order of their appearance in $\mathbb{K}_P$). Note that we ignore all other edges in $\mathbb{K}_P \setminus \mathbb{K}_P^0$, since they do not change the topology with respect to 0-cycles.

Consider only the ordered set of topology changing edges $\mathbb{K}_0 = \{e_1, e_2, \ldots, e_{n-1}\}$ from the above filtration. By definition, the length of these edges satisfies $|e_1| < |e_2| < \ldots < |e_{n-1}|$. While these inequalities might not hold in practice (two consecutive edges could have the same length), a simulated small perturbation [26] of the points can ensure this property holds. The following lemma, which shows the equivalence between $\mathbb{K}_0$ and the Euclidean distance minimum spanning tree (EMST) computed over P, provides the basis for our projection algorithm.

Lemma 2. *Given a set of points $P = \{p_1, p_2, \ldots, p_n\}$, let G be the complete weighted graph defined over P such that the weight of each edge (p_i, p_j) is equal to the Euclidean distance $d(p_i, p_j)$ between the corresponding end points. Then, the ordered set of topology changing edges $\mathbb{K}_0 = \{e_1, e_2, \ldots, e_{n-1}\}$ is precisely the set of edges of the minimum spanning tree (MST), in increasing order of weight, computed over G.*

Proof. We prove this by showing, through induction, that the ordered set of topology changing edges are the same as those added by the Kruskal's algorithm [18].

Consider the first edge e_1 of the filtration. By definition, it is the edge with the smallest length, and thus also the first edge that is added by the Kruskal's algorithm. Let, edges $e_1, e_2, \ldots, e_{i-1}$ be the first $i-1$ edges added by the Kruskal's algorithm. The induction hypothesis is that at this stage, the connected components created by the filtration is exactly the same as the set of connected trees created by the Kruskal's algorithm.

Now, consider the i^{th} topology changing edge of the filtration e_i. For sake of contradiction, say the i^{th} edge added by the Kruskal's algorithm is $e' \neq e_i$. This implies that the edge e' has length less than that of e_i, and connects two connected subtrees together. Then, by definition, e' will occur before e_i in the filtration, and will also be a topology changing edge. Thus, the case of $e' \neq e_i$ is not possible. □

The following proposition follows from the above lemmas and it guarantees the existence of a mapping (projection) that retains the topology with respect to 0-cycles in the projected space.

Proposition 1. *Let P be a set of points in $\mathbb{R}^d$, $\mathbb{K}_0 = \{e_1, e_2, \ldots, e_{n-1}\}$ be the ordered subset of topology changing edges in $\mathbb{K}_P^0$ and C_P^i be the set of connected components obtained during the filtration over $\mathbb{K}_P^0$ after the addition of the first i topology changing edges $\{e_1, e_2, \ldots, e_i\}$. Let $\mathscr{M} : \mathbb{R}^d \to \mathbb{R}^k$ be a mapping that maps points in P to P', and $\mathbb{K}'_0 = \{e'_1, e'_2, \ldots, e'_{n-1}\}$ be the set of topology changing edges in $\mathbb{K}_{P'}^0$. Then, there exists at least one mapping $\mathscr{M}$ satisfying the following properties:*

(a) edge lengths $|e'_i| = |e_i|$, $\forall i \in [0, n-1]$;

(b) the components generated by the filtrations are identical, i.e., $C_{P'}^i = C_P^i$ $\forall i \in [0, n-1]$; and

(c) $PD_{P'}^0 = PD_P^0$, where $PD_{P'}^0$ and PD_P^0 are the persistence diagrams of $\mathbb{K}_{P'}^0$ and $\mathbb{K}_P^0$ respectively.

We abuse notation in the above proposition when stating that $C_{P'}^i = C_P^i$. What this notation means is that the mapping $\mathscr{M}$ establishes a one-to-one relation between the components in C_P^i and $C_{P'}^i$, that is, every point in a component $C' \in C_{P'}^i$ is the image of a point in the corresponding component $C \in C_P^i$. Note that in the above proposition, guaranteeing properties (a) and (b) is a sufficient condition for (c).

The proof of Proposition 1 is constructive and is provided in Sect. 3.3. In fact, using the above lemmas, we design the iterative algorithm shown in Procedure TopoMap that projects a set of high-dimensional points onto $\mathbb{R}^2$ while guaranteeing the properties stated in Propositon 1. The algorithm places the points onto a plane such that the minimum spanning tree edges are preserved. In other words, it "draws" the minimum spanning tree maintaining the edge lengths.

The algorithm initially maintains all the points as a separate component, and stores the minimum spanning tree edges as an ordered set. In each iteration, the algorithm then adds the smallest edge from this ordered set to connect two components, thus reducing the number of maintained components by one. The length of the edge is preserved during this step, that is, its placement is such that distance between the

Procedure TopoMap

Require: High dimensional points $P = \{p_1, p_2, \ldots, p_n\}$

1: Compute the Euclidean minimum spanning tree E_{mst} over P
2: Let $E_{mst} = \{e_1, e_2, \ldots, e_{n-1}\}$ be the edges ordered on length
3: Let $P' = \{p'_1, p'_2, \ldots, p'_n\}$, where $p'_i = (0,0)$, $\forall i$
4: Let $C_i = \{p'_i\}$ be the initial set of components
5: **for** each $i \in [1, n-1]$ **do**
6: Let (p_a, p_b) be the end points of edge e_i
7: Let C_a be the component containing p'_a and C_b be the component containing p'_b
8: Place C_a and C_b in $\mathbb{R}^2$ s.t. $\min_{p'_j \in C_a, p'_k \in C_b} \{d(p'_j, p'_k)\} = \text{length}(e_i)$
9: Let $C' = C_a \bigcup C_b$
10: Remove C_a and C_b from the set of components, and add C' into this set
11: **end for**
12: **return** P'

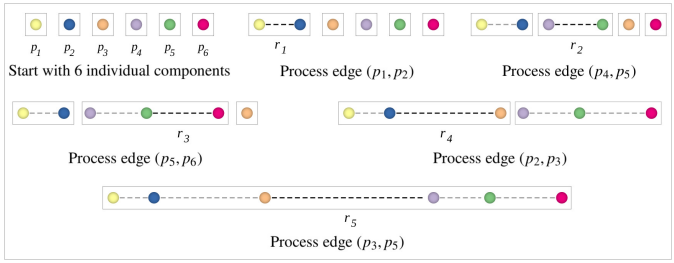

Fig. 3. Projecting the points from Fig. 2 in 1-dimensional space. Each iteration processes one edge (in increasing order to length) from the minimum spanning tree.

two components is equal to the length of the connecting edge that, in turn, has the same length as its counterpart in the original space. This edge is then removed from the edge set and the process repeated until all edges are appropriately placed. The key aspect of the algorithm is Line 8 that places the points based on the minimum spanning tree edge lengths. We describe different ways of accomplishing this in the next section. The maintenance of the set of connected components is accomplished using the union-find data structure [18].

3.3 Building the Topology Preserving Mapping

The TopoMap algorithm starts with e_1, the smallest topology changing edge in $\mathbb{K}_0$. Placing the end points of e_1 (which are individual components at the start of this procedure) in a lower dimensional space such that this distance is preserved is straightforward. In other words, $e_1 = e'_1$. Now, suppose that the first $i-1$ topology changing edges have been added while ensuring Proposition 1. In the i^{th} step, let edge e'_i be added so as to connect two components from $C^{i-1}_{P'}$ that are counterparts of the components in C^{i-1}_P linked by e_i. As mentioned in Line 8 of the algorithm, the goal now is to place these two components such that the minimum distance between them is *equal* to $|e_i|$. If this condition is satisfied, then the properties $|e_j| = |e'_j|$ and $C^i_{P'} = C^i_P$, $\forall j \in \{1, \ldots, i\}$, are naturally attested. By repeating the process for all e_i, $i \in \{1, \ldots, n-1\}$ we also guarantee that $PD^0_{P'} = PD^0_P$. Therefore, proving Proposition 1 now requires showing that there exists a way to place two components connected by an edge e'_i whose length is $|e_j|$. In fact, there are several ways in which this can be accomplished as we show next. Note that, there *always* exists a valid solution to this problem.

A solution in 1-dimensional space. Consider two sets C_1 and C_2, where each point in this set is associated with a x and y value corresponding to its 2D coordinates. Let $p_r \in C_1 | p_r.x > p'.x, \forall p' \neq p_r \in C_1$. In other words, p_r is the *rightmost* point in C_1. Similarly, let $p_l \in C_2 | p_l.x < p'.x, \forall p' \neq p_l \in C_2$ be the *leftmost* point in C_2. Let the input edge length be d. The trivial solution is to simply translate all points in C_1 such that $(p_r.x, p_r.y) = (\frac{-d}{2}, 0)$ and $(p_l.x, p_l.y) = (\frac{+d}{2}, 0)$. Fig. 3 illustrates this procedure for the example shown in Fig. 2.

A geometric 2D solution. Note that the above solution places all the points only along the x-axis. Thus, to obtain a more compact solution that also uses the second dimension, we modify the above solution as follows, by arranging components in the plane with local rotations,

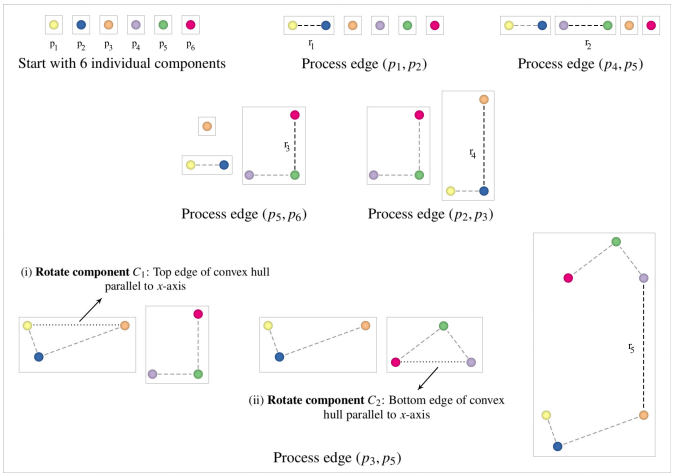

Fig. 4. Projecting the points from Fig. 2 in 2-dimensional space.

similarly to circular layout strategies in tree drawing [67]. Let $hull(C_1)$ and $hull(C_2)$ be the convex hulls of the two components. Pick an edge e_t from $hull(C_1)$ and e_b from $hull(C_2)$. Transform (rotate) C_1 such that e_t is parallel to the x-axis, and is the topmost edge of the convex hull (i.e., has the highest y- coordinate). Similarly, transform C_2 such that e_b is also parallel to the x-axis, but is the bottommost edge. Let $left(e)$ denote the left endpoint of the edge e. Now, translate component C_1 such that $left(e_t) = (0,0)$, and component C_2 such that $left(e_b) = (0,d)$. Alternatively, the right endpoints of the edges e_t and e_b can be used as well to align the two components.

There are different ways in which the edge of the convex hull can be selected (as well as to decide which end point is used for the alignment). We decided to choose the edge that contains one of the end points of the minimum spanning tree edge that is under consideration. In case this is not possible, we choose the edge closest to this point. The intuition here is to not only preserve the connected components after every iteration, but to also try and preserve the end points of the minimum spanning tree edges as much as possible.

Fig. 4 illustrates this procedure for the example points in Fig. 2. Note that the addition of the first two edges result in the same state as in the 1D solution above. However, when the third edge (p_5, p_6) is added, then the point p_6 is placed in a perpendicular orientation. When the last edge (p_3, p_5) is processed, since both components have more than two points each, the convex hull is used to perform the alignment by appropriately transforming both components.

An optimization-based 2D solution. During data analysis, in addition to preserving the topology changing edges of the filtration, it might be beneficial to also possibly preserve other properties as much as possible. In this section, we show how our projection approach can be tuned to support such modifications.

For example, it might be natural to consider a case where we are also interested in keeping the resulting projection "compact", in the sense that we want the points to be as close to each other as possible, while still ensuring that the distance between the two components, C_1 and C_2, is equal to the given value. One way of doing this is to minimize the sum of squared distances between the points in C_1 and C_2 after the placement. This can be achieved using the optimization model described next.

First, fix one of the components, say C_1, and expand it to contain all points in the plane that have distance to C_1 less or equal d (this is the region which *should not contain* any point from C_2). This is achieved by considering lines that are parallel and at a distance d to the edges of $hull(C_1)$. Since this expanded set of lines is also a convex hull, its inner region can be described by a set on linear inequalities on the plane. Let this set of linear inequalities be denoted by $A\mathbf{x} \leq b, A \in \mathbb{R}^{k \times 2}, b \in \mathbb{R}^k$.

The goal then becomes to find the rigid motion (rotation plus translation) that applied to C_2 minimizes the sum of the squared distances to the expanded convex hull without penetrating it. Formally, this problem can be mathematically formulated as:

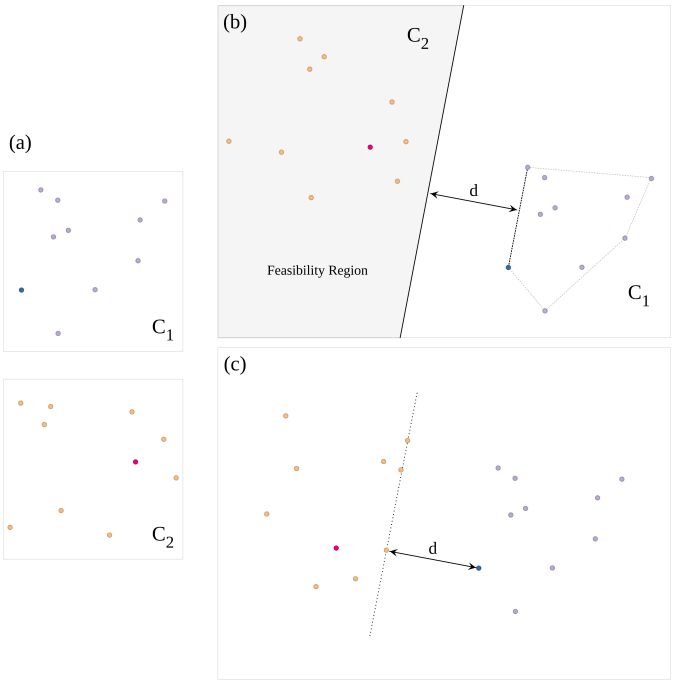

Fig. 5. Solving the optimization model to place two sets of points. **(a)** Two components that are to be merged. **(b)** The feasibility region with respect to the highlighted edge of $hull(C_1)$ is shaded gray. C_2 is initially placed in this region and the optimization is solved. **(c)** Solution.

$$\min_{\theta,t} \sum_{p_1 \in C_1, p_2 \in C_2} \|p_1 - (R(\theta)p_2 + t)\|^2$$
$$\text{s.t.} \quad A(R(\theta)p + t) \text{ is not strictly smaller than } b, \forall p \in C_2.$$

where θ represents an angle with rotation matrix $R(\theta)$ and t is a translation vector. Note that it is possible to consider only the points that define the convex hull of C_2 above.

This problem can be cast as a mixed integer nonlinear optimization problem. The integer variables are needed because the constraints in the above model are actually a "union of sets" instead of an "intersection of the sets" that is usual in optimization. Unfortunately, solving such problems for a large number of points is impractical. We therefore decided to use a simplified heuristic that (i) individually optimizes with respect to each edge of the convex hull $hull(C_1)$; and (ii) minimizes the sum of distances between points in C_2 to a single point in C_1. Formally, let A_i be a row of the matrix A and b_i the respective right-hand side. Let $p' \in C_1$ be a point of interest. We first solve the following optimization problem for $i = 1, \ldots, k$.

$$\min_{\theta,t} \sum_{p \in C_2} \|p' - (R(\theta)p + t)\|^2 \quad \text{s.t.} \quad A_i(R(\theta)p + t) \geq b_i, \forall p \in C_2.$$

We then consider as final solution the one that obtained the smallest objective value. If we also want to preserve the edge from the filtration in the projection, weights can be applied to the objective function above, such that the edge endpoint in C_2 has higher weight when compared to other points. In our implementation, we consider p' to be the end point of the edge that is being processed in that iteration.

Fig. 5 illustrates this optimization process. It shows sets of points C_1 (colored violet) and C_2 (colored orange) that are to be placed at a distance d from each other (Fig. 5(a)). The points colored red and blue correspond to the filtration edge under consideration. The blue point is chosen as the point of interest in order to minimize the objective function of the optimization. The points in C_2 are first randomly placed in the feasibility region corresponding to one of the edges of $hull(C_1)$, and the optimization problem is solved (Fig. 5(b)). The resulting solution is shown in Fig. 5(c).

While the simplified optimization model is also nonlinear (due to the rotation), it does not have integer variables and can then be solved by standard nonlinear optimization algorithms. Note that the final solution is not guaranteed to have two points, one in C_1 and the other in C_2, at exact distance d (and therefore, do not satisfy the required filtration constraint). However, this can still be ensured by sliding C_2 parallel to the edge of $hull(C_1)$ that is associated to the solution obtained in the optimization process.

Table 1. Data sets used in our experiments.

Data set	# Instances	# Attributes	# Classes
Iris [23]	150	5	3
Seeds [23]	210	8	3
Heart [23]	261	11	2
Cancer [23]	699	11	2
Mfeat [23]	2000	64	10
MNIST [45]	20000	784	10
Urban	17520	6	not labeled

Implementation. TopoMap was implemented using C++. It can be divided into two phases. First is to compute the Euclidean distance minimum spanning tree, for which we used the implementation of the dual tree EMST algorithm [48] provided by the *mlpack* library [20], and has a time complexity of $(O(N \log N \alpha(N))$, where N is the size of the input. The next phase is to layout the points. Each iteration of TopoMap aligns two components corresponding to the MST edge being processed. We use the union-find data structure to maintain the list of components, which can be accomplished in $O(N\alpha(N))$ time. The convex hull of the resulting merged component is then computed using the qhull library [4], which takes $O(n \log n)$ time to compute the convex hull of n points. However, since in each iteration, we use only the points in the convex hull of the individual components, $n << N$ in practice. On several large data sets, we found that the layout phase using the geometric approach scaled linearly with the input. On the other hand, computing the EMST became the primary bottleneck when increasing dimensions.

For the optimization based approach, we use the Algencan [61,62] library for solving our optimization model. It is a robust and high performance implementation of the augmented Lagrangian method for nonlinear optimization problems whose code is freely available. As expected, the optimization approach was slower than the geometric approach. However, the main bottleneck was still the EMST phase especially for large point clouds.

3.4 Robustness

From the topological perspective, it is well known that the persistence diagram is robust to noise [17], especially in the context of topology inference [16]: small displacements of points in the original space induce small variations in the persistence diagram. Since TopoMap strictly preserves the persistence diagram, topological robustness to noise is guaranteed by definition. On the other hand, the ordering of the filtration might change slightly due to the noise induced perturbation. Thus, with respect to the actual projection itself, the locations of the points in 2D space might vary marginally when using the geometric solution. Note that the optimization-based approach, being non-deterministic, can produce different layouts even for the same input when run multiple times. However, since the connected components (that represent the points in the persistence diagram) are robust, these components are *always* maintained by the projection.

4 Mapping Evaluation and Interpretation

In this section we present the results of applying TopoMap to project high-dimensional data to a visual space ($\mathbb{R}^2$). Our goal in this section is to analyze the properties of the layout produced by TopoMap, the way it visually encodes the information contained in the high-dimensional data, and how much "readable" the TopoMap layout is when compared to the ones produced by dissimilarity preserving projection methods.

To facilitate the above analyses, we use several data sets (see Table 1) having different numbers of instances and dimensions, some of which are labeled (i.e., the classes are known for the instances). We used the implementations provided by scikit-learn (v0.19.0) for existing methods. All experiments were run on a machine with an Intel(R) Xeon(R) CPU E5-2630 v2 running at 2.60GHz and 64 GB of memory.

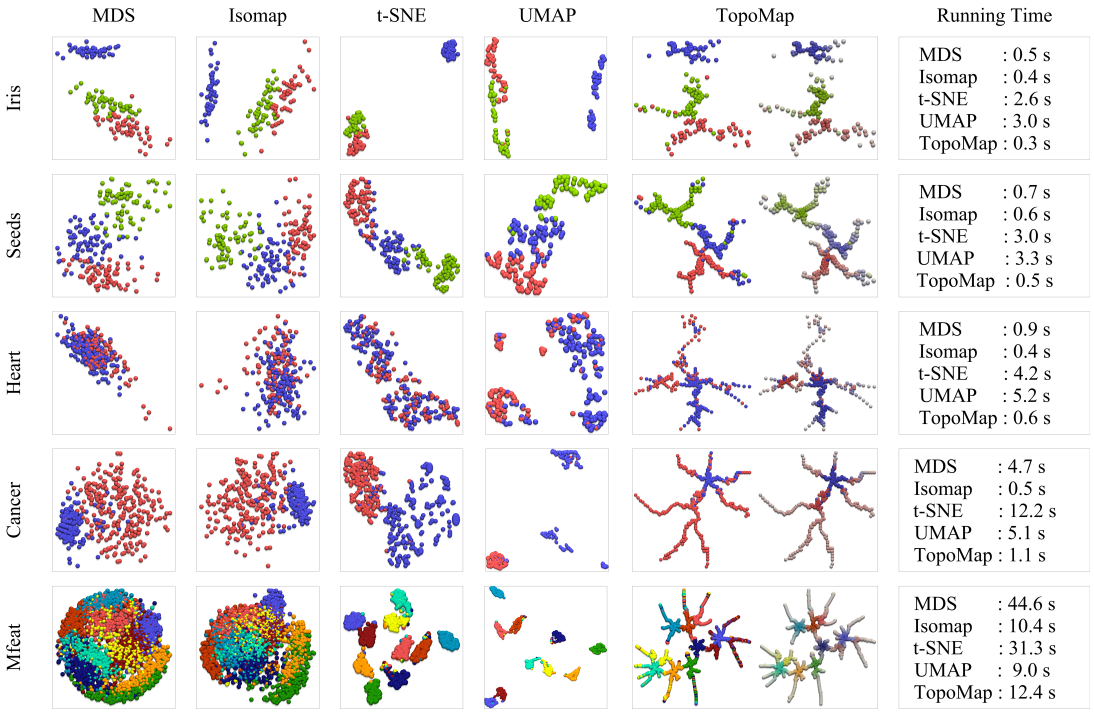

Fig. 6. Layouts produced by MDS, Isomap, t-SNE, UMAP and TopoMap when applied to the five first data sets in Table 1. Right images in the TopoMap column highlight in colors the denser areas in the left images.

4.1 Layout Interpretation

Fig. 1 compares the layout produced by MDS, Isomap, t-SNE, UMAP, and TopoMap for 3 synthetic data sets with well known properties. The first is simply a set of points sampled from three Gaussians, the second is points sampled from three rings, while the third is sampled from two concentric spheres. This figure illustrates the ability of TopoMap to preserve the connected components observed in the original space and to nicely reflect their relative adjacencies. Fig. 6 performs the same comparison with respect to the first five data sets in Table 1. One can notice from these examples that the layouts resulting from TopoMap are quite different from the ones produced by dissimilarity-based methods. This is not a surprise, since TopoMap preserves $n-1$ distances between connected components while dissimilarity-based methods try to preserve n^2 distances (or distributions in the case of t-SNE) between instances.

Star Shaped Ensembles. TopoMap produces a layout made up of star shaped ensembles with branches connecting and emanating from them. One way of interpreting this layout is through the use of the equivalence between the 0-homology filtration of the Rips complex and hierarchical clustering using the single-linkage criterion [13]. The connected components built during the filtration *are exactly the same* as the clusters formed when moving up in the hierarchy. In other words, hierarchical clustering with single-linkage produces, by construction, identical results when considering the input (high-dimensional) data and the two-dimensional projection provided by TopoMap. Thus, when interpreting our projections, users should visually identify centers of stars, as these correspond to clusters in the data (these also tend to correspond to the denser parts of the projection, see the TopoMap results of Fig. 6, right column). On the contrary, the tips of the stars' branches should be interpreted as outliers or points lying at the boundary between clusters (these correspond to the least dense regions of the projections). Notice from Fig. 6 that there is a good correspondence between the star ensembles and the classes of data instances. If using the star shaped ensemble to guide the exploration, TopoMap enables visual analysis that does not demand a great cognitive effort to figure out which are the main groups of instances in the data.

Overall, compared to dissimilarity-based methods, TopoMap is equally, if not more, informative. In fact, except for the Heart data set, one can easily build a visual correspondence between star shaped ensembles and classes. However, even in the Heart, TopoMap indicates the presence of groups of similar instances while the layouts resulting from MDS, Isomap and t-SNE are meaningless. In the Cancer data set, MDS and Isomap clearly reveal one well defined group (blue dots), however, without the labels, it would be difficult to claim that the red points make up a class. The same is true with t-SNE as well, which clearly pinpoints the compact class (in red), but it splits the blue class into a number of local clusters, increasing the potential for misleading interpretation. TopoMap, on the other hand, shows two star shaped ensembles, one well defined and another more elongated, indicating the presence of two classes, one of them not so compact.

Density and Dispersion. The right most images in Fig. 6 (TopoMap column) highlight in colors the denser regions in each layout produced by TopoMap, while gray regions correspond to less dense areas. In particular, we use a Kernel Density Estimator (KDE) with a Gaussian kernel (one Gaussian is centered at each point in 2D and the sum of the contributions of all Gaussians is considered as a density estimation at each point). We additionally use an opacity transfer function, driven by this density estimation, that the users can further adjust if needed (by default, a simple threshold at half of the maximum estimated density). The density-based visualization makes it easier to identify tightly connected groups. Although density-based visualizations have been used to evaluate dissimilarity-based methods [50], the presence of errors and distortions prevent the analysis from being accomplished with high confidence [3]. Note that the "centers" of the starred ensembles correspond to denser areas of the layout, thus corresponding to tightly grouped data instances. This is evident in the examples involving the Heart and Cancer datasets, where a density analysis in the layout resulting from MDS, Isomap, and t-SNE would be of little use.

Branches and Outliers. When considering the above mentioned density based visualization, it is easy to see that branches stemming from the starred ensembles typically are low density regions.

These branches are essentially of two types: those connecting the starred ensembles; and the ones emanating outwards from the stars. The latter is composed of points whose neighborhoods are not tightly connected. From the hierarchical clustering perspective, these can be considered as single point clusters (outliers for example) which merge with already existing large clusters as one moves up the hierarchy.

For example, the TopoMap projection of the Cancer dataset in Fig. 6

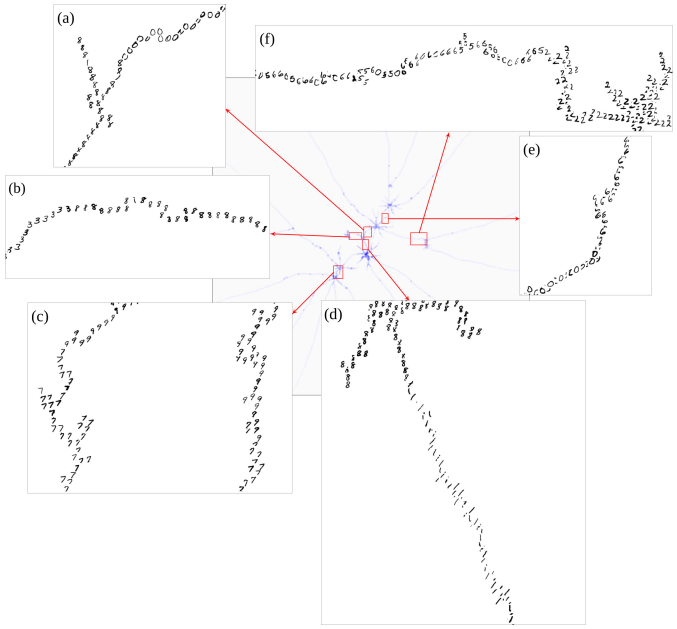

Fig. 7. Mnist data projected using TopoMap (using cosine distance). Transitions between the different starred ensembles clusters: (a) 0 and 8. (b) 3 and 8. (c) 7 and 9. (d) 1 and 8. (e) 0 and 6. (f) class 2 while being a cluster, is far from 0 and is connected to it via outliers.

contains one compact and another more sparse class. The sparse class (red) gives rise to a starred ensemble with a small "center" and long branches emanating from it. The density visualization, coupled with the guarantee that the topology changing edges' lengths are preserved by TopoMap, gives us confidence to claim that the longer branches comes from the sparser class. Moreover, outliers tend also to be part of the loose branches, mainly in less dense areas of the layout. This fact can be observed in the projection of Mfeat dataset, where classes become mixed in loose branches (TopoMap column left image), but not in the center of the ensembles.

Transitioning Between Clusters. Branches connecting the centers of starred ensembles tend to encapsulate instances that lie between clusters, and represent a transition from one group to another. To illustrate this property, we used a uniform sample of 20000 instances from the MNIST data set, and projected it using the angular distance as distance metric (see Sect. 4.4 for a discussion on this). Fig. 7 highlights the different transitions between the star ensembles captured by TopoMap. In particular, note that the cluster corresponding to class **8** transitions to **0** (a), **3** (b), as well as **1** (d). Other transitions such as from class **7** to **9** (c) and from **6** to **0** (e) can also be clearly seen. When well defined clusters are far apart from each other, as is the case of cluster corresponding to class **2** (Fig. 7(f)), the branches emanating from are seen to be formed by "outliers" lying in between the clusters. Notice, however, that the TopoMap layout clearly shows clusters located far apart in the layout, making it easy for users to be aware of which branches are more prone to be made up of outliers (using the density based visualization to help this process).

In general, dissimilarity-based techniques capable of emphasizing clusters such as t-SNE and UMAP does not capture well the transitioning between the clusters. In contrast, techniques capable of grasping transitions, such as Isomap, do not emphasize clusters well. Therefore, besides its theoretical guarantees, TopoMap bears properties difficult to be simultaneously present in dissimilarity based projection methods.

Information loss. There are two main scenarios that can result in a loss of information during the TopoMap projection. First, since the focus is on preserving the 0-cycles, any information with respect to higher dimensional cycles is lost. The 3 rings example in Fig. 1 is one such instance, where the 1-cycles formed by the 3 main components are simply represented as 3 star ensembles. A similar loss can be seen in the concentric spheres example, where the 2-cycles are lost in the projection. Another scenario which can result in incorrect interpretation is when exploring the long branches of the star ensembles. The distance between two points adjacent in a long branch is not necessarily the distance between them in the high dimensional space. Rather, it represents the *distance between the corresponding connected components*. For example, say a point p_1 is connected to p_2 in the minimum spanning tree (and hence is an edge of the filtration). This does not guarantee that p_1 and p_2 will form an edge in the projection—the edge will be between the connected components corresponding to p_1 and p_2. Thus, the two points may be assigned to different branches of a star ensemble depending on the strategy used during the projection. Thus, this property must be considered when interpreting the layout.

Layout interpretation guideline. Based on the above observations, we use the following guideline to explore data using TopoMap for the remainder of this paper:

- Use a density-based colormap to visualize the projection.
- Start exploration by looking at centers of stars with high density. These typically represent clearly distinct clusters in the data.
- Use low density stars to study "uncommon behaviors".
- Explore branches to analyze sparse clusters and outliers.

4.2 Case Study with Unlabeled Data

There are several urban data sets available representing different facets of the city corresponding to its different properties. These are typically studied in isolation, and can sometimes result in missing out on interesting patterns resulting from the interactions between these facets. For example, when analyzing just the count of taxi trips, it is easy to observe that both Times Square as well as Penn Station are identified as hot spots [22] almost throughout the day. Given that the former is a popular tourist attraction, while the latter is a transit hub, one would however expect differences in the way usage patterns of these places change depending on other conditions.

In this experiment, our goal is to see if such patterns do exist. To do so, we generate high dimensional data sets by combining the NYC taxi data and the weather data as follows. We divided two years (2014-2015) into hourly intervals. Then, given a location, we consider the taxi pickups that happened within a 100 m radius of the location. We then create one high dimensional point for every hourly interval having the following dimensions: count of taxi pickups, average fare, average distance, temperature, precipitation (rainfall), and wind speed. Thus, the data set corresponding to each location is a collection of 6D points.

We then projected the data corresponding to Times Square and Penn Station using TopoMap and visually analyzed the patterns present in the projection and analyzed the different stared ensembles by looking at the temporal distribution of the points forming these clusters. The analysis of the data corresponding to Penn Station can be found in the supplemental material.

Times Square. Fig. 8 shows the results obtained for Times Square. In this scenario it is interesting to note that the most dense region (Figs. 8(b) and (c)) correspond to the summer months. This is in turn divided into two clusters: Fig. 8(b) corresponding to main part of the day (10 am to 6 pm), while Fig. 8(c) corresponds to night hours. Similarly the winter months formed its own cluster (Fig. 8(d)). It was interesting to note that there was also a cluster with a smaller number of points corresponding to the Spring and Autumn months (Fig. 8(e)).

A curious observation in the above cases was that none of these clusters included the time period 4 am–8 am. We found these points for summer in a separate cluster shown in Fig. 8(f). On further analysis, we found that this is primarily because the taxi activity at these times was not only lower than the other times, but that these trips also had longer distances and fares than normal. Additionally, we also found that the points corresponding to periods when there was rainfall formed a less dense cluster among the outliers (Fig. 8(g)).

4.3 Probing Dissimilarity-based Projections

Since TopoMap bears theoretical guarantees it can be used to probe other projection methods in order to further understand how those methods behave, specially regarding distortions and cluster preservation. To illustrate this, consider the layouts in Fig. 9. The layout on the top is

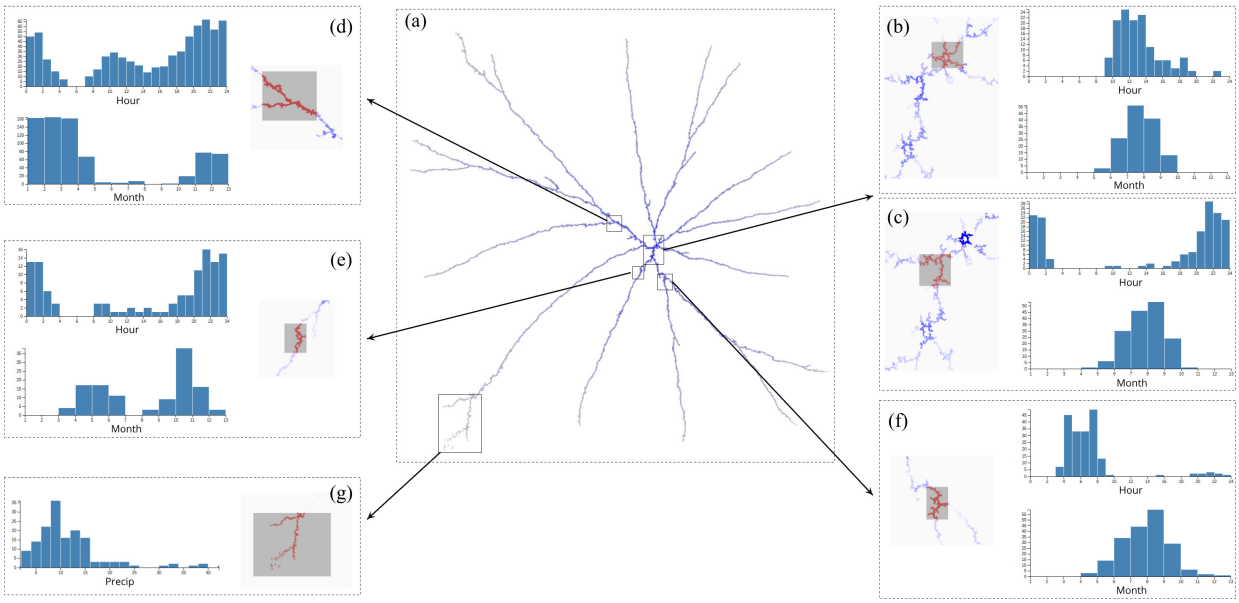

Fig. 8. Analyzing Times Square using TopoMap. **(a)** Projection obtained using TopoMap. **(b)–(g):** Different clusters are selected and the temporal distributions of the selected points visualized as a histogram.

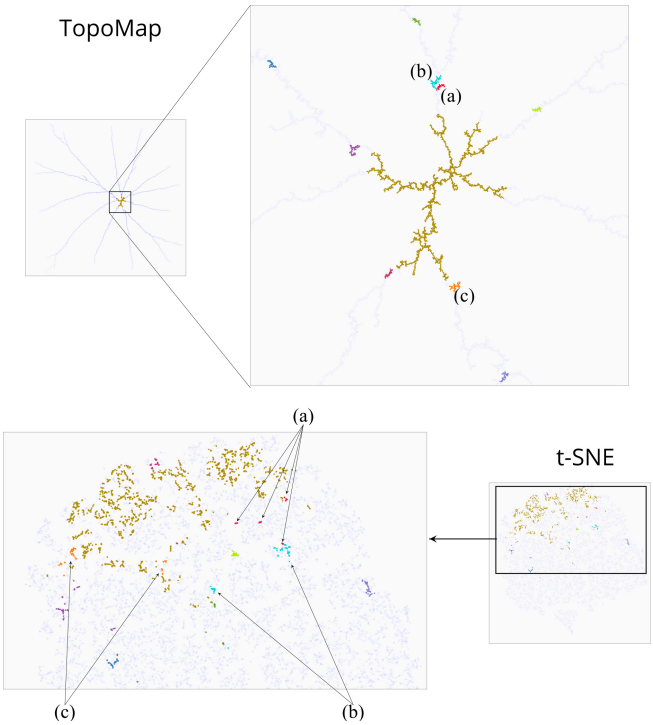

Fig. 9. Connected components (colored groups) guaranteed to exist in the high-dimensional space are broken apart by t-SNE.

the TopoMap projection of the urban data used in the previous section corresponding to Times Square. The highlighted points correspond to the 10 largest connected components obtained by stopping the topological filtration after adding 5000 topology changing edges. There is one large component (gold) and nine smaller ones highlighted in different colors. Recall that if we apply this filtration in the high-dimensional space we would get exactly the same connected components.

The bottom image of Fig. 9 shows the result of projecting the same data using t-SNE. The highlighted points here corresponds to the same components from the TopoMap layout. Notice how t-SNE spreads the large gold component around the layout. Even tightly connected components such as the ones indicated as (a), (b), and (c) in TopoMap layout are broken apart by t-SNE. This example reveals an important property of t-SNE, namely, clusters visualized in a t-SNE layout tend to correspond to pieces of clusters present in the high-dimensional data. With the help of TopoMap, one can realize where t-SNE is placing the different pieces of a cluster. Although experienced users are usually aware about this "breaking cluster" property of t-SNE, we are not aware of any work capable of revealing the extent/intensity of this phenomenon. Revealing this nature of t-SNE is quite important, and can be considered as a side contribution of the present work helping to illustrate the potential of using TopoMap as an analytical tool.

4.4 Discussions

Using alternate distance metrics. As shown in Sect. 4.1 (Fig. 7) TopMap can also be used with an alternate distance metric. This requires computing the MST using this metric, in which case the running time for computing the MST degenerates to $O(N^2)$ due to the computation of the distance matrix. Note that when another distance metric is used, the filtration in the projected space is still preserved with respect to Euclidean distance in the visual space. This also makes it easier for the user to gauge the projection in the visualized space, allowing for a comparison between the effect of using different distance metrics.

Other 2D and 3D solutions. While our approach provides a solution ensuring that the 0-dimensional homology is preserved, there can be other valid solutions as well. Depending on the application, one can also trade-off preserving the persistence of outliers to preserving neighborhoods, or optimizing for a different property. Similarly, it would be interesting to see how the outliers would behave when moving to 3D.

5 Conclusions

In this paper we presented TopoMap, the first planar projection technique that is guaranteed to preserve the homology of 0-cycles of the Rips filtration. Evaluation of our approach using a variety of data sets demonstrated several key properties that are desirable in a visual analytical tool: the layout is easy to understand while its theoretical guarantees provide confidence to the users. In the future, we would like to explore ways in which 1-cycles can be preserved as well in the projection. Analyzing the effectiveness of TopoMap to assist clustering mechanisms is another direction we will pursue.

Acknowledgments. This work was partially supported by the DARPA D3M program; Moore Sloan Data Science Environment at NYU; NSF awards CNS-1229185, CCF-1533564, CNS-1544753, CNS-1730396, CNS-1828576; European Commission grant ERC-2019-COG *"TORI"* (ref. 863464), CNPq-Brazil (303552/2017-4, 304301/2019-1); and the São Paulo Research Foundation (FAPESP) - Brazil (2013/07375-0, 2016/04190-7, 2018/07551-6, 2018/24293-0). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and DARPA.

REFERENCES

[1] H. Adams, A. Tausz, and M. Vejdemo-Johansson. Javaplex: A research software package for persistent (co)homology. In *ICMS*, 2014. https://github.com/appliedtopology/javaplex.

[2] K. Anderson, J. Anderson, S. Palande, and B. Wang. Topological data analysis of functional MRI connectivity in time and space domains. In *MICCAI Workshop on Connectomics in NeuroImaging*, 2018.

[3] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7):1304–1330, 2007.

[4] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469483, Dec. 1996.

[5] U. Bauer. Ripser: efficient computation of vietoris-rips persistence barcodes, Aug. 2019. Preprint.

[6] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. PHAT - persistent homology algorithms toolbox. In *ICMS*, 2014. https://github.com/blazs/phat.

[7] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *NIPS*, pages 177–184, 2004.

[8] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. Topoms: Comprehensive topological exploration for molecular and condensed-matter systems. *J. Comput. Chem.*, 39(16):936–952, 2018.

[9] A. Bock, H. Doraiswamy, A. Summers, and C. T. Silva. Topoangler: Interactive topology-based extraction of fishes. *IEEE Trans. Comp. Graph.*, 24(1):812–821, 2018.

[10] I. Borg and G. P. *Modern Multidimensional Scaling - Theory and Applications*. Springer Series in Statistics, 1997.

[11] P. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. Interactive exploration and analysis of large scale simulations using topology-based data segmentation. *IEEE Trans. Comp. Graph.*, 17(9):1307–1324, 2011.

[12] P. Bubenik and P. Dłotko. A persistence landscapes toolbox for topological statistics. *Symb. Comp.*, 78:91 – 114, 2017. https://www.math.upenn.edu/~dlotko/persistenceLandscape.html.

[13] G. Carlsson. Topology and Data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[14] H. A. Carr, J. Snoeyink, and M. van de Panne. Simplifying Flexible Isosurfaces Using Local Geometric Measures. In *IEEE VIS*, pages 497–504, 2004.

[15] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6), 2013.

[16] F. Chazal and S. Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Symp. on Comp. Geom.*, pages 232–241, 2008.

[17] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Disc. Comput. Geom.*, 37(1):103–120, 2007.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 2001.

[19] Z. Cunninghamn, J. P. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *J. Mach. Learn. Res.*, 16(89):2859–2900, 2015.

[20] R. R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, and S. Zhang. mlpack 3: a fast, flexible machine learning library. *Journal of Open Source Software*, 3:726, 2018.

[21] L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo. Morse complexes for shape segmentation and homological analysis: discrete models and algorithms. *Comput. Graph. Forum*, 34(2):761–785, 2015.

[22] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. T. Silva. Using topological analysis to support event-guided exploration in urban data. *IEEE Trans. Comp. Graph.*, 20(12):2634–2643, 2014.

[23] D. Dua and C. Graff. UCI machine learning repository. https://archive.ics.uci.edu/ml/machine-learning-databases/, 2017.

[24] H. Edelsbrunner and J. Harer. *Computational Topology. An Introduction*. Amer. Math. Society, Jan. 2010.

[25] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. *Disc. Compu. Geom.*, 28(4):511–533, 2002.

[26] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.

[27] B. T. Fasy, J. Kim, F. Lecci, and C. Maria. Introduction to the R package TDA. *CoRR*, abs/1411.1830, 2014. https://cran.r-project.org/web/packages/TDA/index.html.

[28] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 09 1969.

[29] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker. Visual Exploration of High Dimensional Scalar Functions. *IEEE Trans. Comp. Graph.*, 16(6):1271–1280, 2010.

[30] S. Gerber, O. Rbel, P.-T. Bremer, V. Pascucci, and R. T. Whitaker. Morse-male regression. *J. Comput. Graph. Stat.*, 22(1):193–214, 2013.

[31] E. Gomez-Nieto, W. Casaca, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato. Dealing with multiple requirements in geometric arrangements. *IEEE Trans. Comp. Graph.*, 22(3):1223–1235, 2016.

[32] D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing molecular interactions in chemical systems. *IEEE Trans. Comp. Graph.*, 20(12):2476–2485, 2014.

[33] A. Gyulassy, P. Bremer, R. Grout, H. Kolla, J. Chen, and V. Pascucci. Stability of dissipation elements: A case study in combustion. *Comput. Graph. Forum*, 33(3):51–60, 2014.

[34] A. Gyulassy, M. A. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE Trans. Comp. Graph.*, 13(6):1432–1439, 2007.

[35] A. Gyulassy, A. Knoll, K. Lau, B. Wang, P. Bremer, M. Papka, L. A. Curtiss, and V. Pascucci. Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials. *IEEE Trans. Comp. Graph.*, 22(1):916–925, 2016.

[36] W. Harvey and Y. Wang. Topological Landscape Ensembles for Visualization of Scalar-Valued Functions. *Comput. Graph. Forum*, 29:993–1002, 2010.

[37] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. *Comput. Graph. Forum*, 35(3):643–667, 2016.

[38] O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proc. ICML*, page 56, 2004.

[39] P. Joia, D. Coimbra, J. Cuminato, F. Paulovich, and L. Nonato. Local affine multidimensional projection. *IEEE Trans. Comp. Graph.*, 17:2563–2571, 2011.

[40] P. Joia, F. Petronetto, and L. G. Nonato. Uncovering representative groups in multidimensional projections. *Comput. Graph. Forum*, 34(3):281–290, 2015.

[41] J. Kasten, J. Reininghaus, I. Hotz, and H. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE Trans. Comp. Graph.*, 17(12):2080–2087, 2011.

[42] J. Krause, A. Dasgupta, J. Fekete, and E. Bertini. Seekaview: An intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. In M. Hadwiger, R. Maciejewski, and K. Moreland, editors, *IEEE LDAV.*, pages 11–19, 2016.

[43] V. Kurlin. A one-dimensional homologically persistent skeleton of an unstructured point cloud in any metric space. *Comput. Graph. Forum*, 34(5):253–262, 2015.

[44] D. E. Laney, P. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Trans. Comp. Graph.*, 12(5):1053–1060, 2006.

[45] Y. LeCun, C. Cortes, and C. J. Burges. THE MNIST DATABASE of handwritten digits. http://yann.lecun.com/exdb/mnist/, 2020.

[46] J. A. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67:29–53, 2005.

[47] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.

[48] W. B. March, P. Ram, and A. G. Gray. Fast euclidean minimum spanning tree: Algorithm, analysis, and applications. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 603–612. ACM, 2010.

[49] C. Maria, J. Boissonnat, M. Glisse, and M. Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *ICMS*, 2014. http://gudhi.gforge.inria.fr/.

[50] R. Martins, D. Coimbra, R. Minghim, and A. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Comp. & Graph.*, 41:26–42, 2014.

[51] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, Feb. 2018.

[52] D. Morozov. Dionysus. http://www.mrzv.org/software/dionysus, 2010. Accessed: 2016-09-15.

[53] V. Nanda. Perseus, the persistent homology software. http://www.sas.upenn.edu/~vnanda/perseus, 2013. Accessed: 2016-09-15.

[54] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment.

IEEE Trans. Comp. Graph., 25(8):2650–2673, 2019.

[55] P. Oesterling, C. Heine, H. Jänicke, and G. Scheuermann. Visual analysis of high dimensional point clouds using topological landscapes. In *Proc. PacificVis*, pages 113–120, 2010.

[56] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. Visualization of High Dimensional Point Clouds Using their Density Distribution's Topology. *IEEE Trans. Comp. Graph.*, 17(11):1547–1559, 2011.

[57] P. Oesterling, C. Heine, G. H. Weber, and G. Scheuermann. Visualizing nd point clouds as topological landscape profiles to guide local data analysis. *IEEE Trans. Vis. Comput. Graph.*, 19(3):514–526, 2013.

[58] P. Oesterling, G. Scheuermann, S. Teresniak, G. Heyer, S. Koch, T. Ertl, and G. H. Weber. Two-stage framework for a topology-based projection and visualization of classified document collections. In *Proc. IEEE VAST*, pages 91–98, 2010.

[59] M. Olejniczak, A. S. P. Gomes, and J. Tierny. A Topological Data Analysis Perspective on Non-Covalent Interactions in Relativistic Calculations. *Int. J. Quantum Chem.*, 120(8):e26133, 2020.

[60] R. Paul and S. K. Chalup. A study on validating non-linear dimensionality reduction using persistent homology. *Pattern Recognition Letters*, 100:160–166, 2017.

[61] J. M. M. R. Andreani, E. G. Birgin and M. L. Schuverdt. On augmented lagrangian methods with general lower-level constraints. *SIAM Opt.*, 18:1286–1309, 2007.

[62] J. M. M. R. Andreani, E. G. Birgin and M. L. Schuverdt. Augmented lagrangian methods under the constant positive linear dependence constraint qualification. *Math. Prog.*, 111:5–32, 2008.

[63] G. Reeb. Sur les points singuliers dune forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus des séances de l'Académie des Sciences*, 222(847-849):76, 1946.

[64] B. Rieck and H. Leitte. Agreement analysis of quality measures for dimensionality reduction. In *Topological Methods in Data Analysis and Visualization*, pages 103–117. Springer, 2015.

[65] B. Rieck and H. Leitte. Persistent homology for the evaluation of dimensionality reduction schemes. *Comput. Graph. Forum*, 34(3):431–440, 2015.

[66] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[67] A. Rusu. Tree drawing algorithms. In *Handbook of Graph Drawing and Visualization*, 2013.

[68] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Comp. Graph.*, 23(1):241–250, 2017.

[69] N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. P. Bos, and S. Rieder. Felix: A topology based framework for visual exploration of cosmic filaments. *IEEE Trans. Comp. Graph.*, 22(6):1745–1759, 2016. `http://vgl.serc.iisc.ernet.in/felix/index.html`.

[70] V. D. Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *NIPS*, pages 721–728, 2003.

[71] G. Singh, F. Memoli, and G. Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Eurographics Symposium on Point-Based Graphics*, 2007.

[72] M. Soler, M. Petitfrere, G. Darche, M. Plainchault, B. Conche, and J. Tierny. Ranking Viscous Finger Simulations to an Acquired Ground Truth with Topology-Aware Matchings. In *IEEE LDAV.*, pages 62–72, 2019.

[73] T. Sousbie. The persistent cosmic web and its filamentary structure: Theory and implementations. *Royal Astronomical Society*, 414:350 – 383, 06 2011. `http://www2.iap.fr/users/sousbie/web/html/indexd41d.html`.

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

[75] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE Trans. Comp. Graph.*, 24(1):832–842, 2018. `https://topology-tool-kit.github.io/`.

[76] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *J. Mach. Learn. Res.*, 9:2579–2605, 2008.

[77] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. Technical report, Tilburg University, 2007.

[78] G. Weber, P.-T. Bremer, and V. Pascucci. Topological Landscapes: A Terrain Metaphor for Scientific Data. *IEEE Trans. Comp. Graph.*, 13(6):1416–1423, Nov. 2007.

[79] L. Yan, Y. Zhao, P. Rosen, C. Scheidegger, and B. Wang. Homology-preserving dimensionality reduction via manifold landmarking and tearing. *CoRR*, abs/1806.08460, 2018.