Comparing Dimensionality Reduction Methods Using Data Descriptor Landscapes

Bastian Rieck*
Interdisciplinary Center for Scientific Computing
Heidelberg University

Heike Leitte†
TU Kaiserslautern

ABSTRACT

Dimensionality reduction (DR) methods are commonly used in data science to turn high-dimensional data into 2D representations. Since data sets contain different structural features that need to be preserved by this process, there is a multitude of DR methods, each geared towards preserving a separate aspect. This makes choosing a suitable algorithm for a given data set a challenging task. In this paper, we propose a comparative analysis of DR methods based on how well their embedding preserves structural features in the high-dimensional point cloud. To this end, we develop a set of data descriptors that assess local and global structural features of point clouds. These features are computed for the high-D point cloud and the 2D embedding. We then use persistent homology to robustly compare the feature functions. An interactive landscape of the data descriptors, based on their topological differences, permits visually exploring the embeddings and their quality. We demonstrate the utility of our workflow by analysing multiple embeddings of high-dimensional data sets from real-world applications.

Index Terms: Computer Graphics [I.3.6]: Methodology and Techniques—Interaction techniques

1 INTRODUCTION

Dimensionality reduction (DR) methods are an indispensable tool for analysing the high-dimensional data sets that commonly occur in data science. These methods are often used to perform exploratory data analysis, intending to gain knowledge about the underlying data distribution. Data science practitioners often assume that their data are sampled from an unknown manifold M. Algorithms such as principal component analysis aim to "learn" a simple model for M. This model may then not only be used for visualization but also for data compression. The challenges here are twofold: First, it is not clear how to choose a "suitable" DR algorithm because such a choice requires knowledge about M. Some algorithms only work if M is convex, for example. Second, even if a suitable algorithm has been found, most DR algorithms have additional parameters (such as a neighbourhood size) that further affect the results.

There are many different DR quality measures that aim to provide guidance for choosing a DR method. These measures usually judge certain aspects of an embedding, such as errors in distances between the high-dimensional data and the embedded data; see Lee and Verleysen [14] for a survey. While DR quality measures can be used to compare the performance of a single DR method on a data set, they are often optimized for a certain method only and thus may not be readily employed to compare different DR methods with each other. For example, multidimensional scaling (MDS) is optimizing the *stress* measure (the square root of the squared difference between distances in the high-dimensional space and in the low-dimensional space). If we were to use this measure to choose a DR method, methods that embed data into their own coordinate system (thereby changing Euclidean distances by e.g. a constant factor) would inevitably be considered less suitable than MDS. While this problem is circumvented by more general quality measures such as the *co-ranking matrix* [14], the values for different DR methods remain incomparable because a well-defined framework for the comparison is lacking—especially when users are interested in preserving multiple aspects of their data.

In the following, we present a method for approaching the two challenges outlined above. Our method is based on the computation of descriptors on the data. A descriptor is a scalar function that permits the quantification of a single property of the data, such as its density. Given numerous descriptors on a data set, we compare their topological differences (using a well-defined, stable metric) on the original data and on a set of embeddings that are obtained via different DR methods (or the same DR method with varying parameters). In contrast to comparing data descriptors using function space distances such as the L_p-distance, topological behaviour has been shown to be more robust against perturbations and transformations (e.g. translations). Furthermore, especially in a data science context, topological data analysis is known to be able to capture characteristics of a data set that often elude other methods [17]. In summary, our contributions are:

- We present a set of data descriptors that are geared towards analysing multivariate data sets.
- We develop a workflow based on persistent homology for estimating the quality of DR methods by comparing the topology of structural descriptors.
- We provide *data descriptor landscapes*, an easy-to-understand interactive visualization of the quality of DR methods on data sets.
- We demonstrate our method on real-world data sets (with state-of-the-art DR algorithms) and show how it can support users in choosing a suitable DR method.

2 RELATED WORK

Dimensionality reduction methods: DR methods are broadly categorized as either linear or non-linear. Principal component analysis [13, Chapter 8] (PCA), factor analysis [13, Chapter 9] (FA), and multidimensional scaling [13, pp. 706–715] (MDS) are three common linear DR methods. We will now briefly expand on the non-linear methods that we use throughout this paper. Please refer to Gisbrecht and Hammer [11] for more a detailed overview of these non-linear dimensionality reduction methods.

Isomap is an early non-linear DR method and was shown to outperform linear methods on some data sets. It incorporates the idea that the approximation of geodesic distances through neighbourhood graphs can improve the way an algorithm handles the intrinsic geometry of a data set. Locally-linear embedding (LLE) uses a different approach by considering a data set to consist of linear patches at a local level and piecing them together for a global embedding. LLE also employs neighbourhood graphs. Hessian LLE (HLLE) is an improved variant for high-dimensional manifolds at the expense of higher computational costs. A similar idea involves the approx-

*e-mail: bastian.rieck@iwr.uni-heidelberg.de
†e-mail: leitte@cs.uni-kl.de

imation of tangent hyperplanes, as used by the local tangent space alignment (LTSA) algorithm. Finally, stochastic methods can also be employed to achieve better runtime performance: The stochastic proximity embedding (SPE) algorithm of Agrafiotis [2] scales linearly with the size of the input data, instead of quadratically (or even worse).

Dimensionality reduction quality analysis: Lewis et al. [15] showed that non-experts generally disagree when having to choose a suitable DR algorithm, thereby indicating the need for quality metrics and quality analysis. The challenge of choosing a DR method hence remains an active research topic. Sedlmair et al. [21] recently showed that 2D scatterplots are the most useful tool for visually comparing the output of different DR methods, which is why we focus solely on 2D embeddings in this paper. Our method is sufficiently general to support embeddings of arbitrary dimensionality, though. Tatu et al. [23] analysed visual quality metrics under aspects of human perception and discovered that users are looking for structural variations of the data, e.g. its apparent density and the distribution of points. Bertini et al. [3] gave a concise overview and systemization of quality metrics used in high-dimensional data visualization. Our work presented in this paper falls into the "Complex patterns" category because we measure quality by comparing functions on data sets. Pagliosa et al. [16] developed an interactive visualization for multidimensional projections, simplifying their exploration and interpolation. Our approach is more general and not restricted to DR algorithms that are projections in the mathematical sense.

Topological data analysis & data science In a previous publication [19], we showed how to use persistent homology to measure the preservation of a single property (such as the density) on multiple embeddings of a high-dimensional data set. We furthermore analysed the agreement [18] of quality measures on a single embedding in order to identify regions where errors are distributed similarly. The method presented in this paper is capable of comparing *multiple* embeddings under *multiple* aspects. It is meant to be applied at the end of a data science pipeline, aiming to augment the knowledge discovery process in data science. In contrast, other methods employ topological techniques very successfully for preprocessing [1] or even directly as a feature descriptor for machine learning [17].

3 Methods

Given a high-dimensional data set and multiple embeddings obtained by DR methods, we aim to find out which aspects of a data set are preserved by a given embedding. Since the original data set and the embeddings exist in different metric spaces, we cannot compare them directly. We thus propose calculating a set of scalar descriptors on the data and its embeddings. Each data descriptor focuses on a single, well-defined aspect such as density. To permit a stable quantification of the differences between an embedding and the original data set, we compare the topological behaviour of the descriptors using persistent homology. In the next sections, we will first motivate three data descriptors. Following this, we will give a concise exposition of concepts from computational topology and explain how to calculate topological distances using persistent homology. We will end this section with a brief comparison of topological distances and distances in function spaces, showing why topological distances are preferable in the context of data science.

3.1 Data descriptors

The underlying idea of our method is to describe a high-dimensional data set using the behaviour of functions defined on it, i.e. a "fingerprinting" approach. This is inspired by previous work of Biasotti et al. [4], who showed the advantages of using auxiliary functions for shape analysis. We will use k subsequently to refer to the data descriptor neighbourhood size because DR methods use the same variable by tradition. However, when mentioning a DR method and a k-value, it refers to the neighbourhood size of the method and not a data descriptor.

Density: Sedlmair et al. [21] previously showed that humans tend to focus on the variations of density in a scatterplot. We thus want this salient property of the data to be preserved and use the *distance to measure density estimator* introduced by Chazal et al. [7]. Using the k nearest neighbours $\{n_1,\dots,n_k\}$ of a data point x, the descriptor is defined as $f_1(x) = -1/k\sqrt{\sum_{i=1}^{k} \mathrm{d}^2(x,n_i)}$, where $\mathrm{d}(\cdot,\cdot)$ denotes a distance measure such as the Euclidean distance.

Eccentricity: The eccentricity of a data point is a measure of its centrality. Carlsson [5] showed its utility for high-dimensional data analysis. The descriptor is defined as $f_2(x) = 1/n\sum_y \mathrm{d}(x,y)^2$, where y ranges over all n input data points and $\mathrm{d}(\cdot,\cdot)$ again denotes a distance measure. Data points with high values are located more on the periphery of a data set, while a data point that minimizes the equation above may be thought of as a "centre".

Local linearity: Linear structures are a common occurrence in scientific data sets, laying the foundation for the LLE algorithm, for example. We define a new data descriptor that is capable of judging the linearity of a neighbourhood of points. For a point x in the data set and its k nearest neighbours, we build a sample $k \times k$ covariance matrix Σ [13, pp. 121–123]. We now diagonalize Σ, i.e. $\Sigma = PDP^{-1}$, where D is a $k \times k$ diagonal matrix containing the eigenvalues $\{\lambda_1,\dots,\lambda_k\}$ of Σ and the columns of P contain the eigenvectors. Without loss of generality, we assume that $\lambda_1 \geq \cdots \geq \lambda_k$. The quantity $f_3(x) = \lambda_1/(\lambda_1+\cdots+\lambda_k) \in [0,1]$ now measures how much of the variance of the data points is explained if we only use a linear subspace, spanned by a single eigenvector, to describe the data locally. High values indicate that the data are locally linear.

Choosing a neighbourhood parameter k: The density descriptor and the local linearity descriptor both require a neighbourhood parameter k that specifies how many neighbouring points are taken into account. Choosing k requires defining the size of regions that are to be considered significant. This depends on the amount of data points. The data sets analysed in this paper comprise around 1000 points. Here, we choose $k \in [10,20]$, meaning that at least 10–20 points are required for a structure to be considered dense or linear. We calculate the data descriptors for every value of k and use their combined mean in the subsequent analysis. If the amount of input data becomes significantly larger, k needs to be increased. Note that in contrast to the k parameter for DR methods (which also defines the size of neighbourhoods), the data descriptors are much more stable [7]; increasing k only results in smoother distributions of the data descriptor values.

3.2 Persistent homology

Persistent homology is a method for summarizing the behaviour of a scalar function on a data set by its topological features. Topological features arise from the connectivity information of the given function on the data and comprise, for example, connected components (dimension 0), tunnels (dimension 1), and voids (dimension 2). We will first discuss the persistent homology of a 1-dimensional function over $\mathbb{R}$ before we briefly cover persistent homology in higher dimensions. The reader is referred to Edelsbrunner and Harer [10] for a more concise introduction.

1-dimensional data: Given a scalar function $f\colon \mathrm{D} \subseteq \mathbb{R} \to \mathbb{R}$, persistent homology describes the connectivity changes in its sublevel sets, i.e. sets of the form $L_c^-(f,c) = \{x \in \mathrm{D} \mid f(x) \leq c\}$. Starting from the smallest function value of f, we iteratively sweep over all function values and keep track of the number of connected components in the graph of the function. The number of connected components only changes at local extrema. More precisely, a local

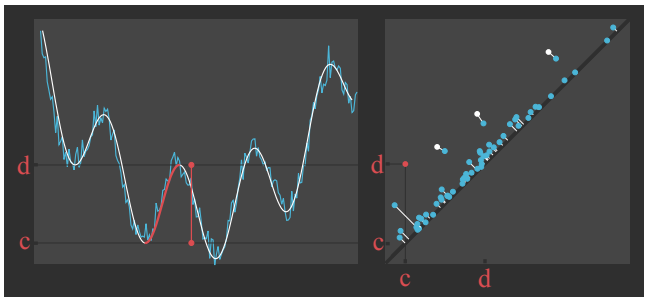

Figure 1: Persistent homology for 1-dimensional data. Each minimum-maximum pair in the data (left) creates a point in the persistence diagram (right). The topological distance between the original function (white) and the perturbed function (blue) is very small.

minimum creates a new connected component, while a local maximum causes two connected components to be merged. We can summarize this information by keeping track of the function values at local extrema. At a local maximum with function value d, we have two connected components belonging to two different local minima with function values c and c', respectively. Without loss of generality, we assume that $c' \leq c$. We now merge the component c into the component c' and store the tuple (c,d) to summarize the merge. The resulting set of tuples in $\mathbb{R}^2$ forms the *persistence diagram* D_f of f (Fig. 1). The quantity $|c-d|$ of a point (c,d) in D_f is called its *persistence*. It is commonly treated as a *significance measure*; large values correspond to stable, significant components that get merged at a very late stage of the sweep, while points near the diagonal are usually seen as an indicator of noise in the data.

High-dimensional data: The example above is a simplified situation because the connectivity of f on its domain D is known implicitly. The lack of connectivity information in high-dimensional data thus necessitates an auxiliary construction. Given a metric such as the Euclidean distance and a distance threshold ε, we calculate the Rips graph R_ε of the data set. This graph has a vertex for every point in the data set and an edge between vertices u and v if the distance between their corresponding points is less than or equal to the distance threshold. The threshold ε controls the approximation of the connectivity information of the data set. There are numerous heuristics for choosing ε [20]. Here, we choose the smallest ε such that R_ε has a single connected component—this accommodates the fact that most scientific data sets do not exhibit well-defined clusters, describing a single manifold structure instead. The advantage of the Rips graph construction is that R_ε contains all Rips graphs $\mathrm{R}_{\varepsilon'}$ for $\varepsilon' \leq \varepsilon$. The threshold ε is thus only a parameter for the *maximum scale*. Information at smaller scales will be included without having to perform additional calculations.

After calculating R_ε, we require a scalar function f, such as a data descriptor from Sec. 3.1, for assigning vertices and edges a scalar value. Each vertex v is assigned the value $f(v)$, while each edge (u,v) is assigned the value of $\max\{f(u),f(v)\}$. This weighting scheme ensures that edges are only added to the graph after both their vertices have been added. To calculate persistent homology, we sort the graph according to its edge and vertex weights and traverse it while keeping track of its connected components using a *union-find data structure* [9, pp. 561–568]. In analogy to the 1-dimensional case, each vertex creates a new connected component, while each edge merges two connected components. This calculation is highly efficient: Given a data set of n points, the persistent homology of R_ε can be calculated in almost linear time, i.e. $\mathrm{O}(n \cdot \alpha(n))$, where $\alpha(n)$ is less than 5 for all practical values of n [9, pp. 573–586].

Higher-dimensional persistent homology: Calculating higher-dimensional topological features such as tunnels and voids requires another auxiliary data structure, the Vietoris-Rips complex V_ε [10, pp. 61–63] of the data set. V_ε is a simplicial complex—a higher-dimensional equivalent of a manifold mesh—consisting of vertices (0-simplices), edges (1-simplices), triangles (2-simplices), and their corresponding generalizations. Briefly put, we obtain V_ε by adding a k-simplex if R_ε contains all of its edges. Analogously to the weight scheme for R_ε, we assign each simplex the maximum weight of its vertices. We now use an algorithm of Edelsbrunner and Harer [10, Chapter VII] that partitions the simplices of V_ε into positive and negative simplices. Similar to the vertices and edges in the description above, positive simplices create a topological feature, while negative simplices destroy one. The result of this process is a set of persistence diagrams for each dimension, summarizing the topological features of the scalar function f on the data set. Calculating topological features up to dimension k can result in a simplicial complex of size $\mathrm{O}(n^k)$ in the worst case. The persistent homology calculation of this complex runs in $\mathrm{O}(m^3)$ time, where m is the size of V_ε. In practice, linear running times are observed. Recent research [22] also points at the possibility of stable linear-size approximations to V_ε, making high-dimensional persistent homology feasible for even larger data sets. For the data sets in this paper, runtime is not an issue yet—comparing 30 different DR methods takes about 7s, which is less than the calculation of a single DR method on the data.

3.3 Function distances in topological spaces

Persistence diagrams are an appealing summary for the behaviour of functions on a data set because they may be compared using well-defined, stable metrics. Given two persistence diagrams X and Y corresponding to functions f and g, we define their *qth Wasserstein distance* as

$$\mathrm{W}_q(X,Y) = \left(\inf_{\eta\colon X \to Y} \sum_{x \in X} \|x - \eta(x)\|_\infty^q \right)^{\frac{1}{q}}, \tag{1}$$

where $\eta : X \to Y$ denotes a bijection between the point sets of X and Y and $\|\cdot\|_\infty$ refers to the maximum distance between two points in $\mathbb{R}^2$. The Wasserstein distance is defined by the minimum amount of displacement (measured as a distance) that is required to transform X into Y. This type of distance measure is very common in e.g. computer vision. Since X and Y usually have different cardinalities, we permit a bijection η to map points from one persistence diagram to their orthogonal projection onto the diagonal, i.e. $(x,y) \mapsto 0.5(x+y,x+y)$. This means that (x,y) has no match in the second persistence diagram.

Fig. 1 illustrates the Wasserstein distance. The blue and white functions are very similar to each other (left). Their persistence diagrams (right) have the same prominent features. Due to the amount of peaks in the perturbed function, there is a large amount of noise near the diagonal. None of these points will be matched to regular points in an optimal bijection. They are all matched to their projections onto the diagonal instead. The total cost of the Wasserstein matching is determined by summing up the lengths of all white edges (with weights according to the q parameter).

The Wasserstein distance between persistence diagrams has excellent stability properties in the presence of perturbations of the data set. A stability theorem by Cohen-Steiner et al. [8] implies $\mathrm{W}_q(X,Y) \leq C^{\frac{1}{q}} \cdot \|f-g\|_\infty^{1-\frac{k}{q}}$, for constants $k \leq q$ and C that depend on f and g as well as on their domain. Here, $\|f-g\|_\infty$ refers to the Hausdorff distance between the two functions. The stability theorem requires that both f and g do not exhibit infinitely many small oscillations that could make $\mathrm{W}_q(X,Y)$ arbitrarily large.

The complexity of W_q only depends on the number of points in both persistence diagrams and not on the sampling resolution or the dimensionality of the input data. In this paper, we will

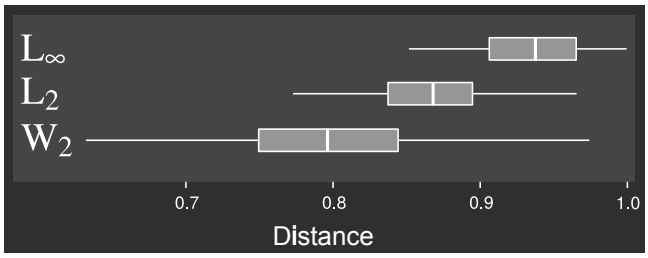

Figure 2: Boxplots of pairwise distances between 1-dimensional functions. Lower values are preferable.

use the second Wasserstein distance W_2 because its local costs are calculated using the common Euclidean distance. Calculating W_q requires finding a maximum weighted matching in a bipartite graph [10, pp. 229–236], which has complexity of $O(m^3)$, with m being the number of points in the larger persistence diagram. The runtime can be drastically reduced by multi-scale approximations of the distance [6].

When calculating higher-dimensional persistent homology of functions f and g as outlined above, we obtain persistence diagrams X_d and Y_d for each dimension d. The qth Wasserstein distance between all persistence diagrams is then defined as the qth root of the sum of all individual Wasserstein distances in each dimension.

3.4 Function distances in metric spaces

In the setting of a metric space, the L_∞ and the L_p (with $p = 2$) distance are commonly used to quantify the similarity between functions f and g. For one-dimensional functions, these distances are easy to compute as

$$d(f,g)_{L_\infty} = \|f-g\|_\infty = \sup_{x\in\mathbb{R}} |f(x)-g(x)| \tag{2}$$

and

$$d(f,g)_{L_p} = \left(\int_{\mathbb{R}} |f(x)-g(x)|dx\right)^{\frac{1}{p}}, \tag{3}$$

respectively. The concept generalizes to higher-dimensional domains $D \subseteq \mathbb{R}^d$. When D is sparse and needs to be approximated, however, costly grid approximations are required. This is also relevant when f and g have different domains D_f and D_g with $D_f \cap D_g \neq \emptyset$. Here, f and g need to be interpolated on both domains, which may quickly become prohibitive in higher dimensions.

To show that the L_∞ and L_p are not suitable for comparing data descriptors, we perturbed the y-values of the function in Fig. 1. We then calculated pairwise distances between the original function and its variants. Fig. 2 shows the corresponding boxplots (after normalization to $[0,1]$ in order to ensure comparability between the different measures). We can see that the mean of the distribution for W_2 is smaller than the means of the function space distances. Hence, W_2 is able to capture similarities between the functions better. Kolmogorov-Smirnov tests between all distance functions confirm that their means are significantly different. Please refer to the supplementary materials for more details.

4 Workflow

In the following, we assume that we are given a high-dimensional unstructured point cloud P and a set of DR methods $\{\phi_1,\ldots,\phi_n\}$. We first calculate the embedding $\phi_i = \phi_i(P)$ for each DR method. We then calculate each data descriptor f_j, $j \in \{1,2,3\}$, on each embedding ϕ_i. This yields a set of function values f_{ij}. After choosing an appropriate value for the distance threshold ε, we calculate the Vietoris-Rips complex V_ε on the point cloud P up to the ambient dimension of the data set. We assign each scalar function f_{ij} as weights to V_ε. The persistent homology of this complex results in a set of persistence diagrams D_{ij} that describe the topological behaviour of the descriptor on the data. We perform the same process (scalar function assignment plus persistent homology calculation) on the original data set, resulting in persistence diagrams D'_j.

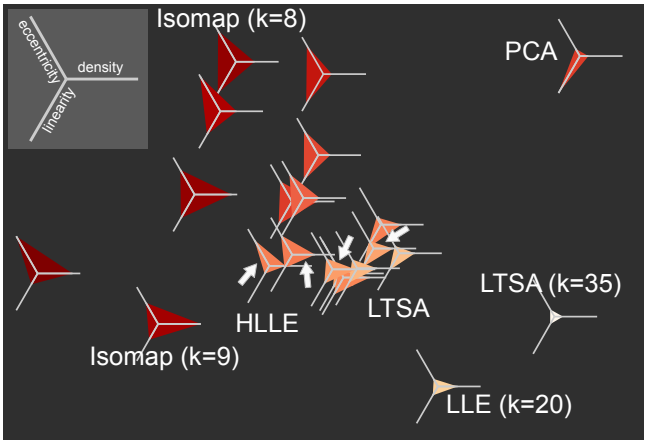

Figure 3: Feature vector landscape for the synthetic faces. The clustering of similar DR methods shows their stability.

For each embedding and each descriptor, we now calculate the Wasserstein distance W_2 between the persistence diagrams D'_j of the original data set and the persistence diagrams D_{ij} of the embedding ϕ_i, $m_{ij} = W_2\left(D'_j, D_{ij}\right)$, and construct an $n \times 3$ matrix M of these distances. Entries in M contain the distance of a given data descriptor (calculated on an embedding) to the same descriptor on the original data. Small distances indicate that the descriptor has a similar topology on the embedding and on the original data. This implies that the property measured by the data descriptor has been preserved well by the embedding.

Each row in M thus measures how well the topology of the data descriptors is retained by the corresponding DR method. We now perform a PCA of M in order to derive coordinates in $\mathbb{R}^2$. In addition, we draw star glyphs [25] to visualize the data descriptor distance values in each row. The glyphs are colour-coded by the Euclidean norm of the row vector, using a sequential colour map that ranges from white to red over orange. A large norm indicates that the row vector contains large values for at least one data descriptor, meaning that the embedding is incapable of preserving its topology properly. This visualization forms the *data descriptor landscape* in which each glyph represents a certain embedding and the distances between glyphs indicate their (topological) similarity.

5 Results

We envision the following strategy for using the information obtained by our method: First, data scientists pick a set of DR methods and apply them to the given data set. If an algorithm has tunable parameters, such as a neighbourhood size, it can be applied multiple times with varied settings. Then, all data descriptors are calculated on all the generated embeddings and the data descriptor landscape is used to select suitable DR methods for embedding the data. This approach is particularly useful when numerous DR methods need to be examined with respect to their parameter stability and the similarity of their embeddings. Especially in case DR is used to reduce the amount of variables in a data set, the data descriptor landscape helps data scientists avoid having to sift through large amounts of auxiliary visualizations such as scatterplot matrices.

5.1 Synthetic faces

This data set was originally described by Tenenbaum et al. [24] as a show-case for non-linear dimensionality reduction methods. It con-

sists of 698 images (64×64 pixels each) that show a 3D model of a human head in different poses and under different lighting conditions. These images are known to lie on a manifold with intrinsic dimension 3, parametrized by two pose variables (left–right, up–down) and one lighting variable. A suitable dimensionality reduction algorithm should reflect these variables in the embedding.

We apply our method to a set of selected DR methods (please refer to the supplementary materials for additional visualizations of the embeddings). All methods except PCA have a neighbourhood parameter k which can be tuned. We vary the values of k and apply our method to the resulting embeddings, using $\varepsilon = 10.5$. The resulting landscape of DR methods is shown in Fig. 3.

We first show how the similarity of embeddings is expressed in the distances within the data descriptor landscape. To this end, we focus on the dense region in the middle of the landscape. It is dominated by instances of HLLE (marked with arrows) and LTSA. Fig. 4 shows some of the corresponding embeddings. Their spatial proximity also indicates that both DR methods are very stable with respect to their neighbourhood parameter. Their glyphs indicate that the largest errors occur in the density and the eccentricity descriptor. Density and distance relations in both types of embeddings are thus not trustworthy. By contrast, the Isomap algorithm is decidedly not stable on this data set—we can see that increasing k from 8 to 9 results in a comparatively large distance in the landscape. The corresponding embeddings (Fig. 4, bottom row) explain this behaviour: Isomap starts to bend significantly when increasing k from 8 to 9. In addition, the colour and shape of the glyphs indicate that Isomap embeddings have comparatively large errors in all descriptors.

We now focus on the outlying glyphs with small norms. They are the embeddings that preserve most of the topological properties of the data descriptors. See Fig. 5 for a comparison. Using the glyphs, we can see that PCA has a high error in the linearity descriptor but is able to preserve density and eccentricity very well. LLE ($k = 20$) features higher errors in the density descriptor, while LTSA ($k = 35$) overall exhibits the least amount of errors in all descriptors. If linear structures and global distances are to be preserved, both of these embeddings are a good choice. If preserving linear structures on a local scale is not important, PCA is the best choice, both in terms of embedding quality and runtime.

5.2 Climate data

Climate researchers often deal with large-scale multivariate data sets—any numeric simulation, which is central to their work, consists of complex models with many variables at increasingly fine resolutions. The aim of these simulations is to provide long-term predictions of changes in world climate. We obtained a large multivariate data set from the *German Climate Computing Centre* (DKRZ). The data set covers a period of one year and is defined over a grid of 192×96 different locations on Earth. It consists of six continuous variables. Each of the 1460 time steps thus contains 18432 vectors in $\mathbb{R}^6$. For the subsequent analysis with $\varepsilon = 2.5$, we will exemplarily average time steps of the meteorological summer season (Jun–Aug) and obtain a random sample of 1000 points. This sampling was chosen because the complete data is prohibitively large for some DR methods.

The data descriptor landscape (Fig. 6) exhibits a clear separation between linear and non-linear DR methods, both in terms of glyph placement, shape, and colour. Except for SPE, all non-linear DR methods are incapable of preserving the data descriptors on this data set. We first focus on the embeddings that are most suitable. FA (with varying number of iterations n), misrepresents all data descriptors to some extent. The misrepresentations decrease with increasing n, though. PCA, on the other hand, somewhat misrepresents density and linearity, while SPE slightly misrepresents both eccentricity and linearity. In contrast to the non-linear methods,

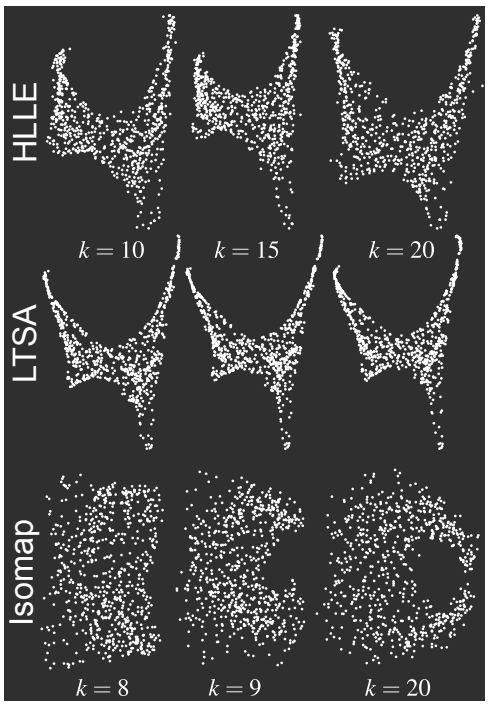

Figure 4: Embeddings generated by HLLE and LTSA resemble each other and stay stable for varying values of k. The Isomap embeddings exhibit very unstable behaviour, by contrast.

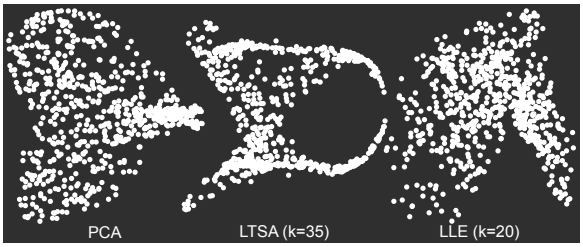

Figure 5: Suitable embeddings for the faces data.

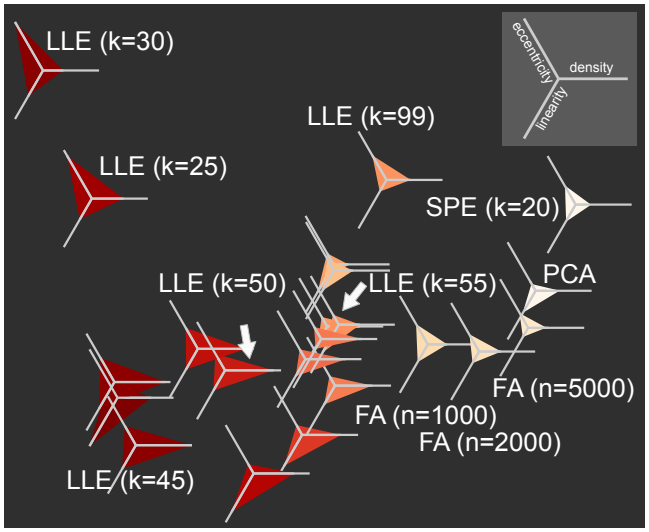

Figure 6: Data descriptor landscape for the climate data. There is a clear separation between linear and non-linear methods.

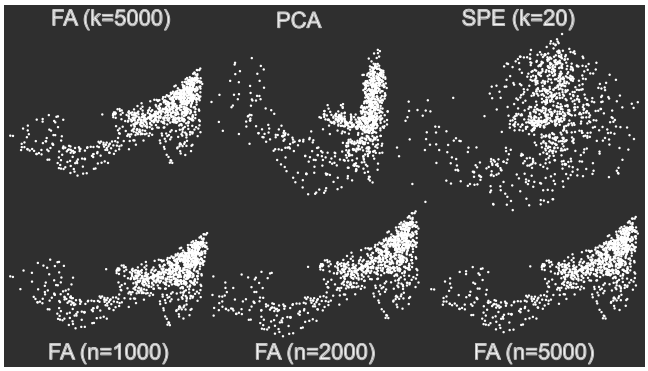

Figure 7: Top row: Suitable embeddings for the climate data set. Bottom row: The effects of tuning the number of iterations for FA.

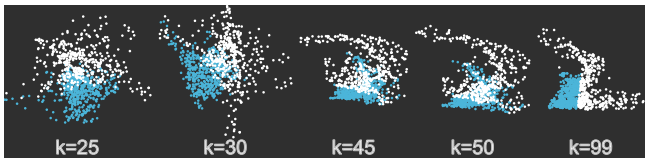

Figure 8: Embeddings generated by LLE are unstable when varying k. Brushing reveals significant differences.

these misrepresentations are relatively small, though. Fig. 7, top row, displays the corresponding embeddings. Through its glyphs, the data descriptor landscape helps us understand that the linear structures that appear to be well-developed in the FA embedding are a salient feature of the data instead of a structural illusion.

The landscape also uncovers the effects of parameter tuning in DR algorithms. LLE, for example, drastically changes form and shape when slowly increasing the number of neighbours used for calculating its linear patches. Fig. 8 shows numerous embeddings. For larger values of k, the separation into a central structure with a connected "flare" is more pronounced. The corresponding glyphs show that neither one of these embeddings is able to preserve eccentricity in the data very well. By contrast, parameter tuning for the FA algorithm is stable, as indicated by the distances in the data descriptor landscape. Here, higher values for the number of iterations n yield very similar embeddings. Fig. 7, bottom row, shows that merely the amount of dispersion changes (in a localized manner) when increasing n.

In conclusion, SPE and FA appear to be the most suitable choices for embedding the climate data, followed by PCA which does not faithfully represent density and linearity when compared to the other methods.

6 Conclusion

We presented a technique for comparing DR methods on a data set. Our visualization supports data scientists in selecting a suitable DR method for working with their data. We used persistent homology to compare the topology of data descriptor functions on a data set and its embeddings. This information was subsequently visualized in the *data descriptor landscape*, a glyph-based scatterplot. We demonstrated the utility of our method on different data sets from real-world applications. Data scientists can use our method to observe the effects of parameter tuning on embeddings and to quickly find similar embeddings without having to display large numbers of auxiliary visualizations. Our method would benefit from being integrated in an established user-centric system for DR methods, such as the *DimStiller* framework by Ingram et al. [12]. We also think that the observed behaviour of some DR algorithms necessitates an investigation of different synthetic and real-world data sets on a larger scale. Data science practitioners are in need of knowing the theoretical and practical limits of their method, as well as the models they are based on to make an informed choice.

References

[1] A. Adcock, D. Rubin, and G. Carlsson. Classification of hepatic lesions using the matching metric. *Computer Vision and Image Understanding*, 121:36–42, 2014.

[2] D. K. Agrafiotis. Stochastic proximity embedding. *Journal of Computational Chemistry*, 24(10):1215–1221, 2003.

[3] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE TVCG*, 17(12):2203–2212, 2011.

[4] S. Biasotti, B. Falcidieno, D. Giorgi, and M. Spagnuolo. *Mathematical tools for shape analysis and description*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool, 2014.

[5] G. Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

[6] A. Cerri, B. Di Fabio, G. Jabłoński, and F. Medri. Comparing shapes through multi-scale approximations of the matching distance. *Computer Vision and Image Understanding*, 121:43–56, 2014.

[7] F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Geometric inference for probability measures. *Found. Comput. Math.*, 11(6):733–751, 2011.

[8] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have L_p-stable persistence. *Found. Comput. Math.*, 10(2):127–139, 2010.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, 2009.

[10] H. Edelsbrunner and J. Harer. *Computational topology: An introduction*. American Mathematical Society, 2010.

[11] A. Gisbrecht and B. Hammer. Data visualization by nonlinear dimensionality reduction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(2):51–73, 2015.

[12] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. DimStiller: Workflows for dimensional analysis and reduction. In *IEEE VAST*, pages 3–10, 2010.

[13] R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. Prentice Hall, 2007.

[14] J. A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72:1431–1443, 2009.

[15] J. M. Lewis, L. van der Maaten, and V. de Sa. A behavioral investigation of dimensionality reduction. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pages 671–676, 2012.

[16] P. Pagliosa, F. V. Paulovich, R. Minghim, H. Levkowitz, and L. G. Nonato. Projection inspector: Assessment and synthesis of multidimensional projections. *Neurocomputing*, 150, Part B:599–610, 2015.

[17] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *IEEE CVPR*, June 2015.

[18] B. Rieck and H. Leitte. Agreement analysis of quality measures for dimensionality reduction. In *Proc. TopoInVis*, 2015. To appear.

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

[20] B. Rieck, H. Mara, and H. Leitte. Multivariate data analysis using persistence-based filtering and topological signatures. *IEEE TVCG*, 18(12):2382–2391, 2012.

[21] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE TVCG*, 19(12):2634–2643, 2013.

[22] D. R. Sheehy. Linear-size approximations to the Vietoris-Rips filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.

[23] A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind. Visual quality metrics and human perception: An initial study on 2d projections of large multidimensional data. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 49–56, 2010.

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

[25] M. O. Ward. Multivariate data glyphs: Principles and practice. In *Handbook of Data Visualization*, pages 179–198. Springer, 2008.