

Agreement analysis of quality measures for dimensionality reduction

Bastian Rieck and Heike Leitte

Abstract High-dimensional data sets commonly occur in various application domains. They are often analysed using dimensionality reduction methods, such as principal component analysis or multidimensional scaling. To ascertain the fidelity of a particular embedding of a data set, users need to analyse its quality. For this purpose, the literature knows numerous quality measures. Most of these measures concentrate on a single aspect, such as preserving the relative distances of points, while others aim to balance multiple aspects, such as intrusions and extrusions in k -neighbourhoods. Faced with multiple quality measures with different ranges and different value distributions, it is challenging to decide which aspects of a data set are preserved best by an embedding. We propose an algorithm based on persistent homology that permits the comparative analysis of different quality measures on a given embedding, regardless of their ranges. Our method ranks quality measures and provides local feedback about which parts of a data set are preserved by an embedding. We demonstrate the use of our technique by analysing quality measures on different embeddings of synthetic and real-world data sets.

1 Introduction

High-dimensional data sets are ubiquitous in most scientific disciplines today. By including more variables, natural phenomena can be modelled and understood more precisely. With an increasing amount of variables, visualization for exploratory data analysis becomes more and more important. A common approach for visualizing complex high-dimensional data employs dimensionality reduction methods: Users have many useful choices available for embedding their data in low dimensions, ranging from linear methods such as *principal component analysis* (PCA) over

Bastian Rieck · Heike Leitte
IWR, Heidelberg University, Germany, e-mail: {bastian.rieck,heike.leitte}@iwr.uni-heidelberg.de

non-linear methods such as *Isomap* to stochastic embeddings such as *t-distributed stochastic neighbour embedding* (t-SNE).

Since ground truth information is often unavailable, quality measures are required to judge how accurately a given method is able to retain a structural property of the data set such as local neighbourhoods. Quality measures usually only assess a single property of the data set. Given an embedding of a data set and several quality measures, users often want to know about compromise solutions: For example, an embedding that distorts local neighbourhoods somewhat but keeps the global structure of the data set intact might be preferable over an embedding that does not distort local neighbourhoods but completely distorts the input data at a global scale.

For an embedding of a high-dimensional data set, we are thus interested in finding out which properties (e.g. neighbourhoods, distances, etc.) of it are faithfully retained. To this end, we analyse the agreement of multiple quality measures on the data. Modelling each quality measure as a scalar field on the embedding, we are interested in the regions of the highest error of a quality measure and hence decompose each scalar field into regions defined by their maxima. Instead of having to compare the quality measures per point, we compare only their decompositions, which are much more stable with respect to noise. If two measures highlight the same regions as having a low quality, their resulting decompositions will be very similar, and we thus consider their behaviour on the data set to be similar. This implies that their respective properties are retained to a similar extent. To measure similarity and highlight areas in which different quality measures disagree the most, we use a similarity measure and a graph matching algorithm. Using several real-world data sets, we demonstrate how our method helps users determine which properties of a data set have been respected by an embedding.

2 Related work

Multi-field data The task of comparing different scalar quality measures on a data set is a particular instance of a *multi-field* problem. In this context, several methods already permit the comparison of scalar functions. Sauber et al. [18] used gradient similarity measures and local correlation coefficients to analyse correlations in scalar fields defined over (regular) 3D or 2D grids. Their approach quickly becomes computationally infeasible with a larger amount of scalar fields. Schneider et al. [19, 20] used *contour trees* for comparing iso-surfaces in two scalar fields. To define similarity between features, they use similarity measures based on the approximated contour volume as well as information-theoretic and graph clustering methods. Their method is geared towards analysing fields for flow visualization and requires cell-based grids, whereas our method is specifically targeting unstructured data.

Scalar field topology and persistent homology Our method uses persistent homology to analyse the topological structure of quality measures on a data set. This

approach is related to other methods from scalar field topology. Gerber et al. [11] use inverse regression in Morse-Smale complexes to obtain a simplified visualization of scalar functions on high-dimensional data. Their method is used for parameter studies but does not permit the comparison of multiple scalar fields. Chazal et al. [4] decompose a scalar field of density values of a manifold into basins of attraction to find stable clusters. Their algorithm does not allow for comparing clusters in different scalar fields but has well-defined stability guarantees that we will use in our approach. Correa et al. [8] use sparse subsets of the Morse-Smale complex to visualize the structure of scalar fields. This approach complements our approach as a visual aid for comparison but does not visualize features in different scalar fields. Oesterling et al. [15] use join trees to visualize the behaviour of density functions on high-dimensional point clouds. Their visualization yields a topological landscape in which regions of similar density may be read off, or a landscape profile that represents clusters as peaks [16]. Both methods are not specifically suited for comparing multiple scalar fields among each other. A complete decomposition of merges and splits of contours in scalar fields is given by the *contour tree* [3]. Our method requires a computation that is similar to the *join tree*, because we are only interested in the maxima of a scalar field.

Dimensionality reduction An in-depth overview of state-of-the-art dimensionality reduction methods is given by van der Maaten et al. [14]. Lee and Verleysen [12] survey numerous quality measures and show how the analysis of global quality measures helps in selecting from a set of different dimensionality reduction methods. Bertini et al. [2] survey quality measures in the context of high-dimensional data visualization. Quality measures are used to provide a “good” overview of a data set. By contrast, our method assumes that the data set has already been embedded and aims on communicating which structural properties are retained with respect to the original data.

3 Quality measures

For the subsequent analysis in Sec. 5, we shall use two groups of local quality measures: Distance-based measures and rank-based measures. The former are more stable against small changes in the embedding whereas the latter are more stable against large changes or linear scaling in the data [12]. In the following, we use pointwise definitions of all quality measures for a data set of cardinality n . We also transform their range such that high values indicate regions of low quality (hence, the functions measure an *error*). Subsequently, d_{ij} refers to the original distances in the high-dimensional space, while δ_{ij} refers to the distances in the embedded space.

Root-mean-square-error (RMSE) RMSE measures the average squared difference between the distances: $f(x_i) = \sqrt{\sum_{j=1}^n (d_{ij} - \delta_{ij})^2 / n}$

Kruskal’s stress In contrast to RMSE, this stress measure penalizes deviations in small distances more than in large distances: $f(x_i) = \sqrt{\sum_{j=1}^n (d_{ij} - \delta_{ij})^2 / \sum_{j=1}^n \delta_{ij}^2}$

Residual variance The residual variance measures the complement of the explained variance between d_{ij} and δ_{ij} , using the linear correlation coefficient:

$$f(x_i) = 1 - R^2(\{d_{i0}, \dots, d_{in}\}, \{\delta_{i0}, \dots, \delta_{in}\}) \quad (1)$$

Spearman’s rank correlation By converting the distances d_{ij} and δ_{ij} to ranks r_{ij} and ρ_{ij} , respectively, this measure is more stable against outliers in the data and invariant to linear scaling: $f(x_i) = 1 - 6 \sum_{j=1}^n (r_{ij} - \rho_{ij}) / (n(n^2 - 1))$

Neighbourhood loss This measure is agnostic to distances and requires an enumeration of the k nearest neighbours of a point both in the original space and the embedded space, which we denote $n_k(i)$ and $v_k(i)$, respectively:

$$f(x_i) = 1 - |n_k(i) \cap v_k(i)|/k \quad (2)$$

Mean relative rank error (MRRE) MRRE measures the mean amount of rank deviations using the k nearest neighbours of the point in both the original space and the embedded space. Lee and Verleysen [12] developed MRRE to penalize two common errors in embeddings, namely very distant points that *intrude* into the k -neighbourhood of a point, as well as very close point that *extrude* from such a neighbourhood.

Note that there are more measures available in literature [12, 14]. We have opted for selecting the most common ones and aimed for those that are not specifically optimized for a certain algorithm. Table 1 gives a short overview of the properties they measure.

4 Methods

In the following, we will describe the components of our method. The main algorithm is similar to the calculation of the *join tree*, but uses persistent homology to

Table 1 Properties that are measured by the quality measures

Measure	Property
RMSE	Average squared distance deviation
Kruskal’s stress	Average squared distance deviation penalizing small deviations
Residual variance	Correlation between original and embedded distances
Rank correlation	Correlation between ranks of original and embedded distances
Neighbourhood loss	Changes in k nearest neighbours; measure of group preservation
MRRE	Extrusions and intrusions of k nearest neighbours

obtain a criterion for the stability of maxima. We refer the reader to Edelsbrunner and Harer [10] for a more detailed account of computational topology and persistent homology.

4.1 Scalar field decomposition using persistent homology

Let \mathbb{D} be a connected domain and $f: \mathbb{D} \rightarrow \mathbb{R}$ a scalar function such as a quality measure. A natural way of summarizing this function for data analysis and comparison is to detect its *peaks* and decompose the data according to the gradient of f , i.e. we decompose \mathbb{D} into disjoint subsets consisting of all those points that reach a certain peak when following the gradient. This approach is also known as *mode-seeking* [5]; see Fig. 1, left, for a simple example. With discrete data, however, mode-seeking approaches are known to be very unstable. To obtain a measure of the stability of the detected peaks, we thus use *persistent homology*, an algorithm from computational topology. Persistent homology summarizes data sets using their topological features. Each topological feature is assigned a significance measure, the *persistence*.

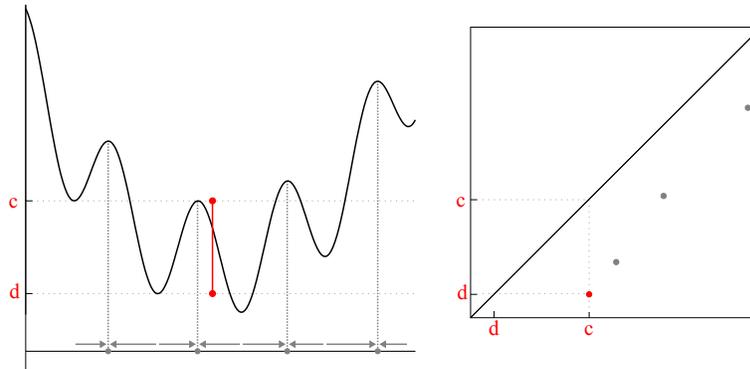


Fig. 1 Left: By following the steepest ascent (shown as arrows on the abscissa), the domain of f is decomposed into disjoint regions. The minima are the boundaries of a region. Right: The persistence diagram of f . The distance from the diagonal is a measure of the stability of a peak.

4.1.1 The 1-dimensional case

Let $\mathbb{D} \subseteq \mathbb{R}$ be our domain and $f: \mathbb{D} \rightarrow \mathbb{R}$ a scalar function. We can use persistent homology to describe connectivity changes in the *superlevel sets* of f , i.e. sets of the form $L_c^+(f, c) = \{x \mid f(x) \geq c\}$ for $c \in \mathbb{R}$. We now traverse the function values of f in decreasing order and keep track of how the connected components of f change.

When we reach a new peak in f , i.e. a maximum of f , a new connected component will be created. By contrast, when we reach a minimum, two connected components are merged into one. We always merge the “younger” connected component (the one with the smaller peak) into the “older” connected component (the one with the larger peak) to ensure consistency [10, p. 150].

By keeping track of the merges, we obtain the *persistence diagram*. It contains a point (c, d) for every connected component created at $c = f(x)$ and merged into an older connected component at $d = f(x')$. Since by definition of the superlevel sets, $c \geq d$, all points in the persistence diagram are located below the diagonal. Fig. 1 illustrates this process for a simple function with several peaks. The *persistence* of a tuple (c, d) is given as $c - d \geq 0$ and serves as a measure of significance. Peaks that quickly get paired with higher peaks result from coarse samplings of a scalar function, whereas peaks with a large difference between creation and destruction may be assumed to represent real features in the data set—in Sec. 4.1.3 we describe an algorithm for automatically finding a significance threshold.

4.1.2 The high-dimensional case

For a discrete set of unstructured points $\mathbb{D} \subseteq \mathbb{R}^n$, we need an auxiliary construction before calculating persistent homology. To obtain neighbourhoods from \mathbb{D} , we use a metric such as the Euclidean distance and a threshold ε to calculate the *Rips graph* \mathcal{R}_ε of \mathbb{D} . The Rips graph has a vertex set of $V = \{0, 1, \dots, |\mathbb{D}|\}$ and an edge set of $E = \{(u, v) \mid d_{uv} \leq \varepsilon\}$, meaning that there is an edge between vertices u and v if their distance (measured using the distance metric) is less than or equal to the selected distance threshold. It endows the unstructured data set with connectivity information, which we require for mode-seeking.

We now apply a decomposition algorithm of Chazal et al. [4] to the scalar field, which proceeds analogously to the 1-dimensional example. The algorithm requires that each vertex v of \mathcal{R}_ε has been assigned its corresponding scalar value $f(v)$ and has a *peak-seeking* and a *merge* phase:

Peak-seeking Traverse the vertices of \mathcal{R}_ε in decreasing order of their function values. Connect each vertex to its neighbour with the largest function value. If the function values of all neighbours are smaller than the one of the current vertex, we have found a peak. The edges that are created by this step correspond to discrete gradient lines of the scalar field in \mathbb{D} . When all vertices have been traversed, we have a collection of tree edges that decompose \mathbb{D} into disjoint regions, similar to the ones shown in Fig. 1.

Merging Traverse the vertices of \mathcal{R}_ε in decreasing order of their function values while maintaining a *union-find data structure* [6, pp. 561–568]. The root of each entry in the data structure corresponds to the peak vertex of a connected component. When arriving at an existing peak during the iteration, a new entry is added to the data structure. Upon arriving at a vertex that is not a peak, we again iterate over all neighbours. We merge neighbours that belong to peaks that are *lower* than the current peak into our component. By contrast, we merge our

current peak with the peaks of all neighbours that belong to *higher* peaks. This changes the value of the current peak, which in turn might trigger other merges with lower peaks again.

This algorithm is a reformulation of the *upper-star filtration* [10, pp. 164–165] in persistent homology. We obtain the corresponding persistence diagram from the algorithm by keeping track of the creation and destruction of components in the merge phase. The peaks of infinite persistence yield the desired decomposition of the domain. In order to have a fine-grained control about which peaks to consider significant and which peaks to prune because they are unstable, Chazal et al. [4] suggest merging peaks based on the differences in their persistence. Given a threshold τ , merges in the second phase of the algorithm are only performed if the peak is lower and the persistence of the peak, i.e. the difference between the peak function value and the function value at the current vertex, is less than the threshold. This procedure results in very stable regions [4]. Since τ affects which peaks are considered relevant and which peaks are considered noise by the algorithm, below we present an algorithm for choosing it automatically, based on the input data.

Choosing ε Varying ε controls the connectivity of unstructured domain \mathbb{D} of the data set. Very small values for ε result in many discrete samples without connections. Very large values for ε , on the other hand, make \mathcal{R}_ε the complete graph on n vertices. Since there is no single “correct” value for ε , several methods have proved to be effective in practice [4, 7, 17]. We use the average distance of points to their k nearest neighbours, for $k \in [10, 20]$, as an initial value for ε [17]. If this procedure does not result in a connected graph, we add edges between the connected components of the graph. These edges are assigned the medoid distance between the connected components.

4.1.3 Threshold selection

Finding a suitable threshold τ involves checking the separation of points in the persistence diagram. A result of Chazal et al. [4, Theorem 4.8] states that relevant peaks can be extracted from the persistence diagram if it contains a band of a certain width (that depends on ε) that does not contain any points. This is the largest empty region we can draw into the persistence diagram (see Fig. 2, middle). The distance to the diagonal from any point within this region is then an admissible value for the threshold parameter τ , which remains stable over a large range.

The theorem makes assumptions about the structure and the sampling conditions of the input data—both of which are unavailable for real-world data. Nonetheless, we can apply a threshold selection process inspired by the theorem. The theorem essentially searches for the largest empty area in a persistence diagram. If this area is deemed large enough (which depends on assumptions about the input data and the function values of the Rips graph), the relevant peaks can be extracted with high probability. We can simulate this decision process by searching for the largest empty area in a persistence diagram and relating its size to the persistence values.

More precisely, we transform the coordinate system of the persistence diagram by a rotation of $\pi/4$, which ensures that the diagonal becomes the new abscissa of the coordinate system. In this transformed coordinate system, we sweep over all points by descending y -value and keep track of the vertical distance between subsequent points. Using the largest vertical distance—which is the width of the desired empty region—and the y -coordinate at which it was detected, we obtain a potential value for the threshold parameter τ . We then calculate the ratio of the width of the largest empty region to the mean width of all empty regions in the persistence diagram. In our experiments, we found that a ratio of at least 4 results in useful and stable thresholds for τ . Smaller ratios are indicative of much noise and may require manual selections via persistence diagrams. In each of our experiments, for instance, automated threshold selection only failed for at most one out of the six quality measures.

4.2 Similarity measure & quality visualization

As a result of the scalar field decomposition algorithm, we are given a set of disjoint regions. We now want to compare the similarity between two such regions over different scalar fields. Given two regions $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_l\}$, where each a and b refers to a vertex in the scalar field, we calculate their similarity using the *Jaccard index*, i.e.

$$J(A, B) = |A \cap B| / |A \cup B| \in [0, 1]. \quad (3)$$

Inspired by the bottleneck distance and Wasserstein distance calculations between persistence diagrams [10, pp. 229–236], we propose an *assignment problem* for assessing the global degree of similarity between two scalar fields. We define the cost between two regions as $1 - J(A, B)$, meaning that we want to penalize regions that do not overlap. To account for different numbers of regions in two scalar fields, we include dummy regions so that a region may also be matched with no region

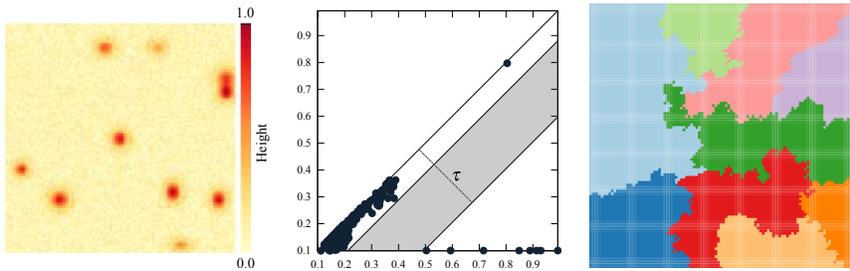


Fig. 2 A noisy scalar field (left), its corresponding persistence diagram (middle) with the region of largest separation, and the resulting decomposition (right). Our algorithm suggests a threshold of $\tau \approx 0.33$. The decomposition remains stable for $\tau \in [0.18, 0.48]$.

from the other scalar field. The total cost of the assignment problem serves as an indicator of how much the decompositions calculated from two scalar fields differ. The pairwise total costs between two scalar fields yields a matrix of pairwise distances. Using a 1-dimensional PCA of this matrix, we obtain a linear ordering of the scalar fields which reflects their respective distances—similar scalar fields are thus placed in proximity to each other. This allows us to read off which properties of an embedding are most likely retained.

Note that the Jaccard index is not capable of differentiating between all functions. It is possible to have two very different functions whose topological decompositions are very similar. We did not encounter this in our experiments, though.

Local similarity scatterplot To provide a more local degree of similarity assessment, we select a reference scalar field. We now solve the assignment problem for each remaining scalar field and keep track of the costs for matching all regions in the reference field. We then visualize the average assignment costs of the reference field using three colours (red, orange, green; each corresponding to 33% of the value range) on the embedding. In the optimal case, all other scalar fields result in the same decomposition as the reference scalar field—the visualization will thus not highlight any region. Green regions indicate an (almost) perfect agreement with all other quality measures. Orange regions show that there are mild differences to the other measures, whereas red regions highlight regions that are severely mismatched with the remaining measures—thereby indicating that a region is unique and does not occur often in the other scalar fields.

5 Results

In the subsequent data sets, we assume that the user has chosen a property that needs to be minimized by the embedding, e.g. stress. With this in mind, users can choose a dimensionality reduction scheme that minimizes this measure on a global scale. We now have the additional task of finding out whether *other* measures are retained as well on the data. As a data pre-processing step, we normalize the scalar values of each quality measure to $[0, 1]$ such that 0 represents no error (highest quality) and 1 represents the maximum error (lowest quality). This is only required to ensure that the scales of different persistence diagrams are more easily comparable among each other.

5.1 *Swiss roll*

The *Swiss roll* data set is a classical data set that was introduced by Tenenbaum et al. [21] as an example of how non-linear embedding methods (*Isomap*) are able to outperform classical linear embeddings (PCA) on certain data sets. The data set con-

sists of a “curled up” plane. Isomap is, by construction, one of the few algorithms capable of embedding this data set properly. We thus work with quality measures on the Isomap embedding. Fig. 3, left, shows the embedding of the data set. We apply our algorithm on the scalar fields induced by the quality measures and choose the threshold automatically. We observe that the distance-based quality measures are defined by exhibiting comparatively large errors along the bottom and the top of the embedded data, while the middle region contains almost no errors. The range of these errors is very small, though. This effect is caused by the unwrapping that distorts distances on a global scale. The same effect occurs somewhat less obviously in the two rank-based quality measures. Here, the impact of the unwrapping is somewhat mitigated by the local neighbourhood size—although we now have the additional error source of small changes in neighbourhoods. Our automated threshold selection results in the same decomposition of the data, though. The measures hence exhibit a very similar behaviour on the data set, proving that Isomap preserves their properties to the same extent. Combined with the knowledge about the small range of the errors, we thus conclude that Isomap yields a perfect embedding of the *Swiss roll*.

5.2 Handwritten digits

We use the *Optical Recognition of Handwritten Digits* data sets from the UCI Machine Learning Repository [1]. The data set consists of 5620 instances of 64-dimensional feature vectors describing the handwritten digits of multiple writers. We will compare the behaviour of different quality measures on a linear embedding (PCA) and on a non-linear embedding (t-SNE) of the data.

PCA All quality measures on this data are rated similarly by our algorithm. We thus omit the visualization of the relative distances. Fig. 4 shows a selection of different quality measures for this data set. All exhibit a large spread in their value range, indicating that the errors are substantial. Except for MRRE, all measures have a single region of varying size in the middle of the data set. MRRE, on the

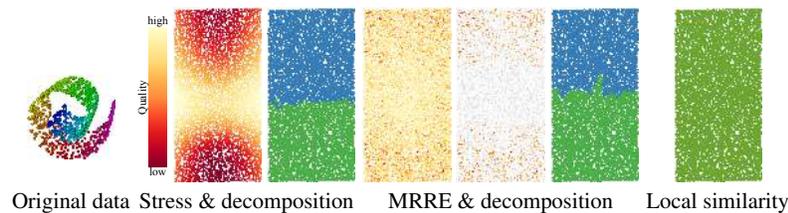


Fig. 3 Swiss roll and two example measures with their decompositions. The distribution of maxima in MRRE is not apparent in the scatterplot. Excluding values below 30% of the range displays this behaviour better. Our decomposition picks up on the fact that the error is maximal in the bottom and top region of the embedding, just as for the other quality measures.

other hand, seems to contain multiple smaller peaks but not a single expressive region. Our decomposition shows that due to the higher baseline error in the quality measure, the detected peaks do not have a sufficiently high persistence. Thus, the decomposition of all measures results in a single large region. By assigning labels to the data set (Fig. 4, bottom), we see that they have their pronounced maximum around the region of the digits “5”, “8”, and “9”, as this region is not well-separated by PCA. PCA for this data set is thus a typical example of a compromise solution: The embedding contains high errors in all quality measures (i.e. its overall quality is rather low), but the errors accumulate in a single known region in all quality measure.

t-SNE The relative distances of the different quality measures (Fig. 5, top) show that three measures (MRRE, rank correlation, and residual variance) are very similar: With a high baseline error, they only have one significant peak in each of the regions of the embedding. For the neighbourhood loss measure, we find only one significant peak in the marked region, i.e. it does not separate into two smaller sub-regions. The stress measure yields a more fine-grained decomposition. Here, regions A, B, C, and D are split (instead of staying a larger region). Region B, for example, corresponds to a separation of the digit “1”. The higher stress values in the bottom part of this region indicate that the distances in t-SNE do not reflect the high-dimensional distances very well. RMSE differs the most from the remaining measures. Here, we see an additional split introduced by two new regions A and B. This is caused by two peaks of high persistence at the top and bottom of the

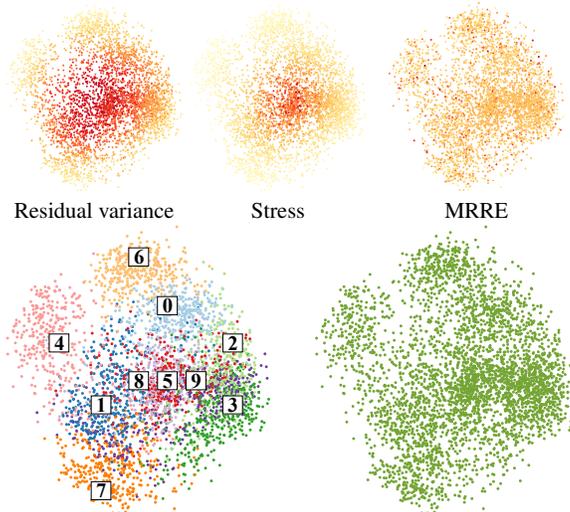


Fig. 4 Errors in the PCA projection concentrate on a single region in the data. This is readily seen in the plot of residual variance and Stress, for example, but not in MRRE. Here, larger errors seem to be distributed uniformly. Upon decomposing the data, these peaks are shown to be of low persistence. The best behaviour for modelling the MRRE is thus a *single* peak in the data.

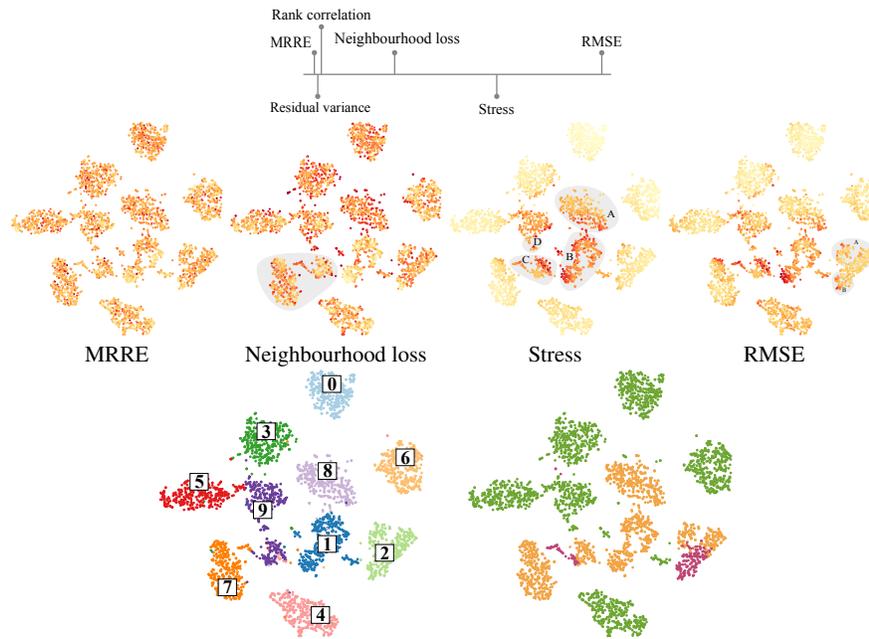


Fig. 5 Large errors in the t-SNE projection occur uniformly in the regions corresponding to the digits. This is indicated well by MRRE, for example. RMSE, however, highlights regions with a decidedly non-uniform error distribution.

marked region. The new regions correspond to the digit “2”. RMSE highlights that t-SNE is distorting the distances for these points in favour of the (very good) global separation in the embedding.

The contested regions are clearly highlighted in the similarity scatterplot (Fig. 5, bottom), using MRRE as a reference measure. The orange and red regions are the regions in which the other measures differ the most from MRRE: For the most part, t-SNE is able to separate the data set very well, but local distances are somewhat distorted. The distortions are very localized and concern only some digits, though. When comparing the PCA and the t-SNE embedding, we can compare the residual variances, for example, to conclude that t-SNE commits smaller distance errors on average, thereby offering a better global separation of the digits. Hence, t-SNE might be preferable over a PCA embedding, despite the localized errors it introduces.

5.3 Concrete compressive strength

The *concrete compressive strength* data from the UCI Machine Learning Repository [1] contains 1030 mixtures of 8 different concrete compounds. The data are

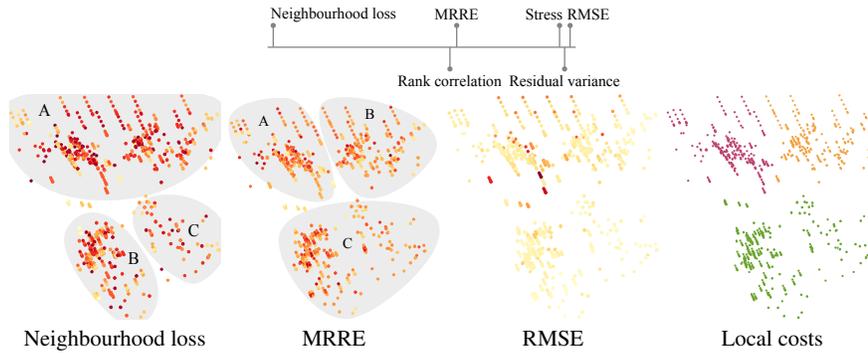


Fig. 6 Neighbourhood loss exhibits three maxima of high persistence, resulting in a decomposition in three regions. MRRE yields a different decomposition in three regions. RMSE as a representative of distance-based measures exhibits large errors in the upper region only. Using neighbourhood loss as a reference measure, the local similarity scatterplot shows that the measures disagree mostly in the upper region of the embedding.

known to exhibit linear structures [11, 13]. Previous research [13] indicates that PCA is sufficient to provide an overview of the data. The relative distances (Fig. 6, top) indicate that there are three groups of quality measures on the data. The distance-based quality measures stress, RMSE, and residual variance all exhibit a single large peak around the top of the data set. Since there are no maxima of high persistence in the remaining part of the data set, it will not be decomposed further, resulting in a single large region. MRRE and rank correlation, by contrast, decompose the top part of the data further into two regions A and B because the corresponding peaks are separated by lower values. Neighbourhood loss results in a complementary decomposition, showing that the top part of the data is dominated by a single peak of high persistence, while the bottom part decomposes into two regions B and C.

The local similarity scatterplot indicates the agreement of quality measures with respect to neighbourhood loss. We can see that the measures disagree on both the upper regions, indicating that their error distributions are very different. This leads us to question the quality of the embedding at these areas—which are rated very differently by the rank-based measures, while the distance-based measures mostly agree.

6 Conclusion

We introduced a method for comparing the behaviour of different quality measures for dimensionality reduction algorithms. Our method currently decomposes scalar fields according to their maxima only. For future work, we plan on evaluating whether the inclusion of minima would further increase the expressive power. We

also want to evaluate the effects of different neighbourhood graph type approximations [7]. Furthermore, we want to investigate whether different measures for feature relevance in scalar fields, such as *topological saliency* [9], might further improve the results. Last, we want to look for alternatives to the Jaccard index for ascertaining the similarity of decompositions.

References

1. Bache, K., Lichman, M.: UCI machine learning repository (2013). URL <http://archive.ics.uci.edu/ml>
2. Bertini, E., Tatu, A., Keim, D.: Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE TVCG* **17**(12), 2203–2212 (2011)
3. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. *Computational Geometry* **24**(2), 75–94 (2003)
4. Chazal, F., Guibas, L.J., Oudot, S.Y., Skraba, P.: Persistence-based clustering in Riemannian manifolds. *JACM* **60**(6), 41:1–41:38 (2013)
5. Cheng, Y.: Mean shift, mode seeking, and clustering. *IEEE TPAMI* **17**(8), 790–799 (1995)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT press (2009)
7. Correa, C., Lindstrom, P.: Towards robust topology of sparsely sampled data. *IEEE TVCG* **17**(12), 1852–1861 (2011)
8. Correa, C., Lindstrom, P., Bremer, P.T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE TVCG* **17**(12), 1842–1851 (2011)
9. Doraiswamy, H., Shivashankar, N., Natarajan, V., Wang, Y.: Topological saliency. *Computers & Graphics* **37**(7), 787–799 (2013)
10. Edelsbrunner, H., Harer, J.: *Computational topology: An introduction*. AMS (2010)
11. Gerber, S., Bremer, P.T., Pascucci, V., Whitaker, R.: Visual exploration of high dimensional scalar functions. *IEEE TVCG* **16**(6), 1271–1280 (2010)
12. Lee, J.A., Verleysen, M.: Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing* **72**(7–9), 1431–1443 (2009)
13. Lee, J.H., McDonnell, K.T., Zelenyuk, A., Imre, D., Mueller, K.: A structure-based distance metric for high-dimensional space exploration with multidimensional scaling. *IEEE TVCG* **20**(3), 351–364 (2014)
14. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: *Dimensionality reduction: A comparative review*. Tech. Rep. 005, Tilburg University (2009)
15. Oesterling, P., Heine, C., Jänicke, H., Scheuermann, G., Heyer, G.: Visualization of high-dimensional point clouds using their density distribution’s topology. *IEEE TVCG* **17**(11), 1547–1559 (2011)
16. Oesterling, P., Heine, C., Weber, G.H., Scheuermann, G.: Visualizing nD point clouds as topological landscape profiles to guide local data analysis. *IEEE TVCG* **19**(3), 514–526 (2013)
17. Rieck, B., Mara, H., Leitte, H.: Multivariate data analysis using persistence-based filtering and topological signatures. *IEEE TVCG* **18**(12), 2382–2391 (2012)
18. Sauber, N., Theisel, H., Seidel, H.P.: Multifield-Graphs: An approach to visualizing correlations in multifield scalar data. *IEEE TVCG* **12**(5), 917–924 (2006)
19. Schneider, D., Heine, C., Carr, H., Scheuermann, G.: Interactive comparison of multifield scalar data based on largest contours. *CAGD* **30**(6), 521–528 (2013)
20. Schneider, D., Wiebel, A., Carr, H., Hlawitschka, M., Scheuermann, G.: Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE TVCG* **14**(6), 1475–1482 (2008)
21. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)