EUROGRAPHICS 2025 / A. Bousseau and A. Dai
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 2

Towards Scaling-Invariant Projections for Data Visualization

Joel Dierkes, Daniel Stelter, Christian Rössl, and Holger Theisel

Visual Computing Group, University of Magdeburg

Abstract

Finding projections of multidimensional data domains to the 2D screen space is a well-known problem. Multidimensional data often comes with the property that the dimensions are measured in different physical units, which renders the ratio between dimensions, i.e., their scale, arbitrary. The result of common projections, like PCA, t-SNE, or MDS, depends on this ratio, i.e., these projections are variant to scaling. This results in an undesired subjective view of the data, and thus, their projection. Simple solutions like normalization of each dimension are widely used, but do not always give high-quality results. We propose to visually analyze the space of all scalings and to find optimal scalings w.r.t. the quality of the visualization. For this, we evaluate different quality criteria on scatter plots. Given a quality criterion, our approach finds scalings that yield good visualizations with little to no user input using numerical optimization. Simultaneously, our method results in a scaling invariant projection, proposing an objective view to the projected data. We show for several examples that such an optimal scaling can significantly improve the visualization quality.

CCS Concepts

• ***Human-centered computing*** → *Visualization techniques; Visualization systems and tools;*

1. Introduction

Despite the vast progress of Machine Learning techniques, visual exploration by humans is still an inevitable tool for data analysis. Mainly, this includes the identification of patterns and inherent structure, which allow for reasoning about the data and the process of their generation. This poses significant challenges: First, the visualization must be able to efficiently process and convey data that appear in vast amounts, i.e., there is a huge number of records. Second, we typically experience multivariate data with many dimensions, i.e., each record stores a multitude of different attributes. In this work, we focus on the second problem: The visual analysis of high-dimensional multivariate data ultimately requires the synthesis of a 2-dimensional picture. This means that a *projection* from the high-dimensional data space to screen space is required.

There exists a variety of methods for the projection from high-dimensional data spaces to low-dimensional ones. This is often referred to as dimensionality reduction. Typical examples include the Principal Component Analysis (PCA) [Hot33] and metric Multidimensional Scaling (MDS) [Tor52]. Dimensionality reduction algorithms compute projections that are in some sense optimal for the given input data. PCA, for instance, maximizes the variances in the direction of the orthogonal principal components, which define the projection to the low dimensional space. MDS, in contrast, minimizes the distortion of distances (*strain*) under the projection.

For the input data, each dimension represents some attribute, which is often described by quantities or physical units. Typical examples are element counts, lengths measured in meters, or weights measured in kilograms. Here, data points like $(10, 0.2\,\mathrm{m}, 1\mathrm{kg})$ and $(3, 0.05\,\mathrm{m}, 0.5\mathrm{kg})$ live in a common 3-dimensional space. However, it is unclear how one can relate two points, e.g., what is the *distance* between these two points? Any metric or scaling seems arbitrary unless all attributes represent semantically equivalent phenomena. This is often not the case, even if attributes share the same physical quantity or unit. For instance, extents like width and length can be measured in meters, but it makes probably no sense to compare them with a shipping distance. In summary, any choice of metric for the high-dimensional space or scaling of the given dimensions is typically arbitrary in practice. We also remark that it is difficult to even use the concept of a vector space as there is no semantic interpretation of linear combinations of data points.

The scale of dimensions is arbitrary and can be varied. Unfortunately, the variation of scaling affects the projection to low-dimensional space, which depends on the scaled data. This effect is often significant and results in a high variation of the generated visualizations. Thus, scaling can improve or deteriorate the visual analysis by exposing or hiding features of interest. This holds for most dimensionality reduction method as they are variant to scaling.

The goal of this paper is to use appropriate scaling to explicitly *improve* projections of multidimensional data. We treat the scaling of the input data as a degree of freedom: its variation generates different projection results. We aim at finding an optimal scaling in the sense that certain interesting features in the data become exposed in the 2-dimensional visualization. The idea is to compute

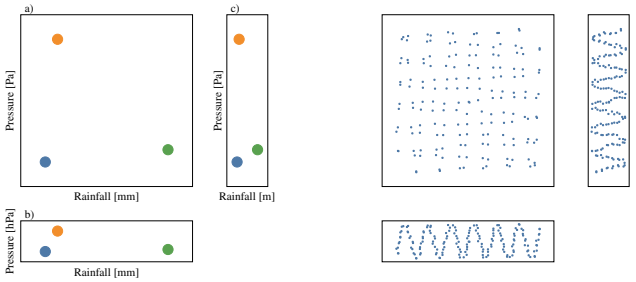

Figure 1: *Scaling dependence. Left: Three data points for which pressure and rainfall are measured in different units. Depending on the chosen units, different clusters occur. (a) three (or one) clusters for pascal/millimeter. (b) two clusters for pascal/meter. (c) two different clusters for hectopascal/millimeter. Right: Different scalings of the axes give different interpretations of the* doublesine *dataset.*

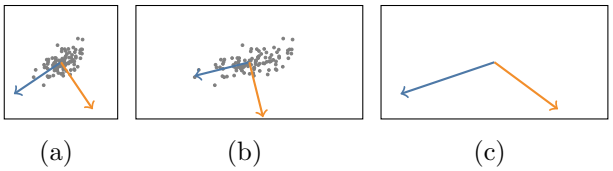

Figure 2: *(a) Original data points (gray) and eigenvectors determined by the PCA. (b) New PCA applied to data points scaled by* 2 *in x_1-direction. (c) Scaling the eigenvectors of the original PCA in (a) by the same factor does* not *yield the eigenvectors in (b): The PCA is* not scaling invariant.

a visually optimal projection for the given data w.r.t. a selected visualization quality metric. This is an automatic process that uses numerical optimization which explores the search space of different scalings. The optimal scaling is invariant to the arbitrary initial scaling of the given input data, and therefore, we can consider the projection method applied to the optimally scaled data as scaling invariant. We apply the three projections PCA, t-SNE, and MDS which are pairwise combined with four different quality measures.

2. Introductory Examples

In this section, we provide some simple examples which emphasize the effect of scaling the input data on the 2-dimensional visualization as output.

Scaling 2D data changes interpretation. We first consider 2D data, i.e., these examples do *not* employ dimensionality reduction. Assume we are given weather data such that each row contains one data point with two values for pressure and rainfall. We can measure pressure, e.g., in pascal (Pa) or in hectopascal/millibar (hPa), and similarly, the rainfall could be measured by a rain gauge in meters (m) or millimeters (mm). Depending on the choice of unit – and therefore the scaling of the pressure and rainfall axis – we obtain different visualizations. Figure 1 (left) shows an example for three data points. Note how the distribution of the data points in 2D screen space varies. Subsets of points that are located close to each other but separated from other subsets appear as *clusters* in the scatter plot. It becomes immediately obvious that the different scales affect the number, location and separation of clusters. At the same time, clusters efficiently convey information about different classes of data points within the dataset. Assume that the colors of the points are an additional label attribute. In this case, one may prefer the scatter plot *(a)* which separates the different classes (here: colors) best. The visual separation of classes is typically important for the interpretation of the data. This example shows that scaling can change the interpretation.

Even if a classification by clusters is not possible or not interesting for a specific application, the following example shows that scaling may change the reasoning about a dataset completely. Figure 1 (right) shows three visualizations of the same dataset (*doublesine*): 168 points in 2D are plotted with differently scaled axes. The square scaling is determined by a Min-Max normalization, resulting in a scatter plot for which the points seem to be aligned on a regular grid. The scales in the x_1 and x_2 axes are reduced in the plots on the right and below, respectively. The scatter plots show two different wave patterns: the first wave runs horizontally while the second wave runs vertically.

Projection methods depend on scaling. We give an example to illustrate how varying the scaling of the input can affect projection methods. In this example, we show this for PCA which builds on principal components. These are the orthogonal eigenvectors of the mean-centered covariance matrix of the data. For dimensionality reduction, the two eigenvectors corresponding to the largest eigenvalues define a 2D basis for projection. The rationale is that the variance of the data is maximized in these directions, i.e., in some sense, the projection incorporates the most significant information encoded in the data.

Figure 2 visualizes a simple experiment where the PCA is applied to a set of data points $\mathbf{x}_i \in \mathbb{R}^2$ and to $\mathrm{diag}(2, 1)\mathbf{x}_i$. The principal components computed for the scaled data differ from scaling the original principal components. This becomes visually most obvious by the loss of orthogonality due to scaling. Consequently, the output of the PCA depends on the scaling of the input data, or vice versa: The PCA is *not scaling invaraint*. We remark that this holds similarly for alternative dimensionality reduction techniques.

Scaling changes visual structures under projection. The final example shows projections of high-dimensional data. The *Penguins* dataset in Figure 3 shows different projections by PCA with different scalings, which was chosen by common scaling techniques as indicated by the bars. A variation of the scaling leads to different projections, which exhibit different structures in the scatter plots. This scale dependence holds also for other, nonlinear projection techniques. Nonlinear projections like t-SNE and MDS similarly depend on the scaling of the input data.

3. Related Work

In this section, we review related work, in particular, projections from high-dimensional spaces, the evaluation of their quality for visualization, and approaches to achieve scale invariance.

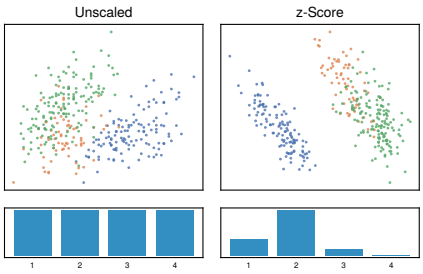

Figure 3: *PCA applied to two different scalings of the* Penguins *dataset: the unscaled (*Left*) and z-Score scaled (*Right*) one. The bar chart visualizes the scales for each dimension. The scatter plots show the 2D PCA projection.*

3.1. Dimensionality reduction and projection

Finding "good" projections from points in high-dimensional data spaces to the 2D screen is a standard problem in different areas of data analysis. There exists a variety of techniques with different taxonomies, and an exhaustive coverage is beyond the scope of this paper, see, e.g., the surveys [NA18, BSL*08, OKMM15, VDMPVdH09, WL10]. Instead, we briefly review a selection of techniques, which are representative in the sense that they facilitate the introduction and classification of our objectives.

Automatic methods. These techniques aim to automatically find optimal projections by linear or nonlinear optimization. Linear techniques minimize a quadratic form, either by solving a linear system or an eigenvalue problem. Examples are the Principal Component Analysis (PCA) [Hot33], Fischer's Linear Discriminant Analysis (LDA) [Fis36], Metric Multidimensional Scaling (MDS) [Tor52], and a variety of their variants (e.g., uncertainty-aware PCA [GSS*20]). Examples of nonlinear techniques are Locally Linear Embedding (LLE) [RS00], Maximum Variance Enfolding (MVU) [WS06], Local Similarity Preserving (LSP) [PNML08], Local Affine Multidimensional Projection (LAMP) [JCC*11], Stochastic Neighbor Embedding (SNE) [HR03], t-SNE [vdMH08], and Uniform Manifold Approximation and Projection (UMAP) [MH18]. Each of these methods comes with several extensions and variants.

Besides the classification into linear and nonlinear methods, there exist further taxonomies of automatic projections [NA18]. In contrast to *non-supervised* methods, *supervised* ones utilize class label information to construct the mapping: They aim to place instances of the same class close to each other in the low-dimensional space. A typical example is LDA. Other methods can be modified to become supervised, such as NeRV [VPN*10], t-SNE, MVU, Isomap [TdSL00], LLE, and LSP.

Automatic projections can be classified into *single-level* and *multi-level*. Multi-level projections typically employ hierarchies and operate in two steps by first building a hierarchical representation of the data, and then mapping data from hierarchical levels to the screen space. Examples of techniques that can be made multi-level are LSP, SNE, LAMP, and Part-Linear Multidimensional Projection (PLMP) [PSN10].

Visual support for automatic projections. The methods mentioned above are automatic, i.e., they work without user interaction. In recent years there emerged an alternative focus of introducing visual interaction with – or guidance of – the projection algorithms. Moreover, visual techniques are used to analyze the results of automatic projection methods. This combination of visual and automatic techniques is driven by Visual Analytics, where a variety of theoretical models and frameworks have been proposed [CBP09, EHM*11, KKE10, SSZ*16, SSS*14].

Several guiding scenarios have been identified [SZS*17] for the interaction with automatic projection techniques: data selection (e.g., [JFSK15]), annotation and labeling [BNHL14, GRM10, HS07], data manipulation [JZF*09], feature selection [JZF*09, ML14], parameter tuning [CLKP10, SBVLK09, Dri12, ML14], constraints definition [EHM*11, BNHL14, DKM06], and type selection [RL15, LWBP14].

Interactive projections. Interactive projections aim to find good projections within a manual or semi-automatic process. The user interactively varies projection parameters such as coordinate axes or anchor points for each dimension. Each variation provides instant visual feedback such that the user "manually" explores the parameter space and the "quality" of a specific parameter configuration is evaluated visually.

iPCA [JZF*09] is an example of a general map, which allows for interactive visual data exploration using the PCA. It lets the user scale the data dimensions prior to PCA projection and visualization by a scatter plot, parallel coordinates, and a correlation matrix. In addition, the system emphasizes the transitions caused by scale variation by fading out the previous visualizations. Our new method is similar to the iPCA in that both methods scale the input data and both apply PCA for projection. However, our goal is not an interactive visual exploration of the data, but instead a one-shot method that provides a single, unique, in some sense optimal projection.

3.2. Criteria for good projections

Several criteria for the quality of a projection have been identified [NA18]: minimizing a certain energy function, excluding or minimizing missing neighbors, excluding or minimizing false neighbors, reproducing clusters in data space in the projection, describing paths, discovering outliers, or evaluation of class compactness. Also, the error function to be minimized may describe a certain measure of distortion, such as Euclidian distance [BSL*08, Kru64, Sam69], Spearmans correlation [SC88], or the Kullback-Leibler divergence [HR03].

Sedlmair and Aupetit [SA15] evaluate different measures that assess the quality of scatter plots. The input is a set of projected 2D data points and associated labels that indicates the membership in a cluster. The output is a scalar quality rating which indicates how well the scatter plot conveys structure in the data including the shape and separation of clusters. Our method applies three of the best quality measures in a numerical optimization. Section 4 provides a more detailed review.

3.3. Scaling invariant projections

There exist well-known standard approaches in statistics to eliminate scaling as a degree of freedom (see, e.g., [Flu97]): *Whitening*

scales data dimensions such that the covariance matrix becomes approximately the identity matrix. This, however, obstructs dimensionality reduction as, e.g., for the PCA the different magnitudes of covariance provide a measure for information content and define the projection. *Min-max scaling* applies an affine transformation to each dimension such that all data values are in the interval $[0,1]$. *Z-score normalization*, also known as *standardization*, scales variances of all features to 1.

Dimensionality reduction depends on the notion of distance in the high-dimensional space. Another option is to change the metric which can be interpreted as a generalized scaling. For example, the Mahalanobis distance [Mah36] adapts locally to the data distribution and is scaling invariant. Finally, there exist sophisticated methods in statistics for selected projections like PCA, e.g.the Copula Component Analysis (COCA) [HL14]. Purely statistical methods, however, do not take into account data visualization as an ultimate goal, whereas our approach evaluates the visual effect of scaling.

Besides these standard approaches, there are multiple "non-trivial" solutions for the scaling problem. Lehmann and Theisel [LT18] present a solution of the scaling problem to linear projection induced by star coordinates. They apply constrained Lloyd relaxation [Llo82] to the projected points, aiming to find the best scaling such that the structures in the projected points are as regular as possible. This way, it is assumed that the perceived structure is in the data and not due to unfavorable scaling. Another idea to achieve scaling invariance is to exclusively consider ordinal data: Only the order relation for each dimension is preserved, rather than to retain the position or distances in Euclidean space. Examples are [Kru64, DLH80]. While these techniques are trivially scaling invariant, the limitation to order relations also limits the applicability.

3.4. Further 2D Scaling Optimizations

Besides projections of data, there are further approaches concerned with scalings of 2D data. In this case, the scaling problem is equivalent to finding an optimal aspect ratio in scatter plots. Fink et al. [FHSW13] propose to optimize the aspect ratio towards getting properties of the Delaunay triangulation of the projected points, e.g., to maximize the minimum angle of the triangles, or to minimize the square sum of the angles of the triangles. Talbot et al. [TGH11] minimize the arc-length of a certain representation in the plot.

4. Quality measures

We aim to find an optimal scaling by maximizing the quality of scatter plots of the 2D projection. Sedlmair et al. [SA15] compare multiple quality measures based on their respective predictive performance on labeled data. While all of them are principally suitable for our method, we select the three measures which achieve high ranking scores. We also formalize a trivial fourth measure. This section briefly introduces and reviews the four quality measures.

For each algorithm, the input is a set of 2D points that are partitioned into classes, i.e., each is associated with a *label*, such that equally labeled points are assumed to be part of the same cluster. In the following, $\mathbf{x}_1 \dots \mathbf{x}_p \in \mathbb{R}^2$ denote the p 2D input points. Furthermore, there are M distinct labels, i.e., a set of classes $\mathscr{C} = \{c_1, \dots, c_M\}$. For each class c_i, let C_i denote the set of indices of all points in c_i, and $|C_i|$ is the number of points in c_i. Let $class(\mathbf{x}_i)$ denote the class of point $\mathbf{x}_i$.

4.1. Distance Consistency (DSC)

The *distance consistency* (DSC), introduced by [MSH09], computes the score of separation of different clusters based on the class membership and distance to the nearest class center. For each point $\mathbf{x}_i$ where the closest cluster centroid differs from its assigned one, the DSC value gets a penalty.

$$\mathrm{DSC} = \frac{\sum_{i=1}^{p} \mathrm{CD}(\mathbf{x}_i, class(\mathbf{x}_i))}{p}$$

with the centroid distance

$$\mathrm{CD}(\mathbf{x}, c_j) = \begin{cases} 0, & d(\mathbf{x}, centr(c_j)) < d(\mathbf{x}, centr(c_k)) \\ & \forall k : 1 \le k \le M; k \ne j \\ 1, & \text{otherwise,} \end{cases}$$

where d denotes a distance metric and $centr(c)$ is the centroid of a cluster c. A lower score represents a better separation of the classes. The computational cost of the DSC is $O(pM)$.

4.2. Class Density Measure (CDM)

The *class density measure* (CDM) was proposed by Tatu et al. [TAE*09]. It evaluates the separation of different clusters for the given input point set. For this, it sums the distances between all pairs of points that belong to different clusters:

$$\mathrm{CDM} = \sum_{k=1}^{M} \sum_{i \in C_k} \sum_{j \notin C_k} \left|\left| p_i - p_j \right|\right|$$

A higher output score indicates better separation of clusters, and thus, a better configuration. The CMD score is invariant to rigid transformation, i.e., rotation and translation, of the input $\mathbf{p}_i$. The computational cost of CDM is $O(p^2)$. In contrast to [TAE*09], we do not employ an additional discretization by rendering a density image, i.e., $\mathbf{p}_i$ are sample coordinates, not pixels.

4.3. Distribution Consistency (DC)

The *distribution consistency* (DC), introduced by [MSH09], evaluates the distribution of data points using gridded histograms. The separation properties of the input points w.r.t. their labels are measured by binning the points and computing the entropy per bin. The 1D version creates bins over the x-axis, the 2D version additionally over the y-axis.

The x and y-axes are partitioned into equally sized intervals of size $(h,o) \in \mathbb{R}^+ \times \mathbb{R}^+$. These are the histogram bins. For the sake of simplicity, we choose a number h and o such that the number of partitions of the x and y-axis is equal. Let p_j^c denote the number of points of class c in bin j, and $p_j = \sum_c p_j^c$ is the total number of points in this bin. Then

$$H(j) := -\sum_{c \in \mathscr{C}} \frac{p_j^c}{p_j} \log_2 \frac{p_j^c}{p_j}$$

is the entropy, which measures the average information content of the bin i. $H(j) = 0$ holds if bin j has only points of one class, and

$H(j) = \log_2 M$ if it contains equally many points of all classes. DC sums the entropy over all bins and applies normalization such that the output is within 0 and 100, which are worst and best results, respectively:

$$\mathrm{DC} = 100 - \tfrac{1}{Z} \sum_j p_j H(j) \quad \text{with} \quad \tfrac{1}{Z} = \tfrac{100}{\log_2 M \sum_j p_j} \,.$$

DC is invariant to rigit transformations of the input points $\mathbf{p}_i$.

4.4. Data Space Ratio (DSR)

The *data space ratio* (DSR) captures the essence that points in a scatter plot should use the available space and not be concentrated in one region. The measure creates a grid over the plot with cell size $(h,o) \in \mathbb{R}^+ \times \mathbb{R}^+$ and counts the number of grid cells that are non-empty. The value is normalized afterwards onto a range of $[0,100]$. p_j denotes the number of points in bin j, n the total number of bins. The measure is determined by:

$$\mathrm{DSR} \;=\; 100 * \frac{\sum_j sgn(p_j)}{n}.$$

Since the DSR does not rely on labels it is also suitable for unsupervised data. For fixed values of h and o the runtime is O(p).

5. Problem statement

As shown before, the result of a projection method depends on the scaling of the n dimensions. We consider the p data points as a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The scaling of the dimensions is given by a vector $\mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{R}^n$ of scales $k_i \neq 0$. Scaling the rows of $\mathbf{X}$ is achieved by left-multiplication with the diagonal matrix $\mathbf{K} := \mathrm{diag}(\mathbf{k}) \in \mathbb{R}^{n \times n}$ to obtain

$$\mathbf{X}' \;=\; \mathbf{K}\mathbf{X}\,.$$

Assume that each dimension (or each row, likewise) represents a physical quantity such as length or weight in a certain physical unit. The quantities and their units are independent of each other and possibly different. As mentioned in the examples, there may be no canonical and/or semantically meaningful way to compare values in different dimensions. Thus, the scales of the dimensions is arbitrary. We interpret this scaling as a degree freedom that should be eliminated.

Finding a *scaling invariant* projection requires to show that the projection of $\mathbf{X}'$ is independent of the choice of $\mathbf{k}$. This means that for any $\mathbf{k}$ the projected data differ only by an isometric transformation (rotation, translation) and possibly uniform scaling. We propose to find an optimal scaling function of the initial data. This means that, independently of the *initial* scaling, the exact same *transformed* data are generated. Thus, the result of a projection method becomes independent of the initial data scaling, too.

Definition 1 (Scaling Invariance) A projection is *scale invariant* if there exists an "optimal" scaling function $\hat{\mathbf{K}}(\mathbf{X}) := \mathrm{diag}(\hat{\mathbf{k}}(\mathbf{X}))$ of the input data such that

$$\hat{\mathbf{K}}(\mathbf{X}')\,\mathbf{X}' \;=\; \hat{\mathbf{K}}(\mathbf{X})\,\mathbf{X}$$

holds true for any data $\mathbf{X}$ and any scaling $\mathbf{K}$ with $\mathbf{X}' = \mathbf{K}\mathbf{X}$.

Thus, our approach eliminates the scaling of the input data as a degree of freedom. However, $\hat{\mathbf{k}}$ is not uniquely determined – we require additional constraints to effectively fix the scaling. This is achieved by selecting and maximizing an objective function that measures the quality of the scaling. For our purpose, this quality is assessed from visual appearance of the projected data.

6. Method

In this section, we construct the following algorithm to find an optimal scaling that determines our scaling invariant projection:

Objective. The input consists in a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ with data points in columns, and each data point is labeled to belong to one of the classes $c_1, \ldots, c_M$ represented as a vector $\mathbf{c} \in \{1, \ldots, M\}^p$. The output is an optimal scaling $\hat{\mathbf{k}}$ which determines the scaling invariant projection.

In order to find $\hat{\mathbf{k}}$, we construct the following objective function, which is then maximized: Let $\mathbf{U} = \mathrm{p}(\mathbf{X})$ denote the projection to 2D points represented as columns of $\mathbf{U}$. Furthermore, let $\mathrm{quality}(\mathbf{P}, \mathbf{c})$ denote the evaluation of a scalar quality measure $\mathbf{c}$ on 2D points $\mathbf{P}$. We define the objective function $f(\mathbf{X}, \mathbf{c}) \;=\; \mathrm{quality}(\mathrm{nrm}(\mathrm{p}(\mathbf{X})), \mathbf{c})$, where $\mathbf{P} = \mathrm{nrm}(\mathbf{U})$ scales the projected 2D coordinates to the unit square such that the values in each dimension span the maximum range of $[0,1]$. The rationale of this 2D scaling is first to mimic the appearance of typical scatter plots without scale information, and second to effectively constrain the range of quality scores, and this way, bound the objective. Note that nrm is applied *after* the PCA projection: The PCA is affected by the data scaling $\mathbf{k}$ but not by the normalization for quality assessment.

We maximize the objective to obtain the optimal scaling $\hat{\mathbf{k}}$ as

$$\hat{\mathbf{k}} \;=\; \arg\max_{\mathbf{k}} f\left(\mathrm{diag}(\mathbf{k})\,\mathbf{X}, \mathbf{c}\right) \,,$$

where the matrix product $\mathrm{diag}(\mathbf{k})\,\mathbf{X}$ scales the rows of $\mathbf{X}$, i.e. the input points (see Section 5).

Numerical optimization. For the numerical optimization, we apply the derivative-free, deterministic Nelder-Mead algorithm [NM65] with efficiency improvements [GH12]. This standard algorithm worked well for our examples. We use the implementation in the Julia package `Optim` [MR18].

The multivariate search space is huge, and the objective function is non-convex and expected to have local maxima while the true global maximum is unknown. Therefore, the numerical optimization is likely to find only a local maximum. This is true for any numerical optimization algorithm, and in particular for the limited budget of optimization steps or function evaluations, which is limited in practice.

While the Nelder-Mead consistently yielded "good" (i.e., probably close to optimal) results in our experiments, the returned scaling $\hat{\mathbf{k}}$ often depends on the initial guess. Even for similar quality values, the scalings that determine these values can differ significantly. To avoid this kind of "noise", we could either run the algorithm for a large number of random initial configurations, or define a canonical initial guess that determines the result. As the first option doesn't provide a guarantee, we favor the second one. More specifically, we

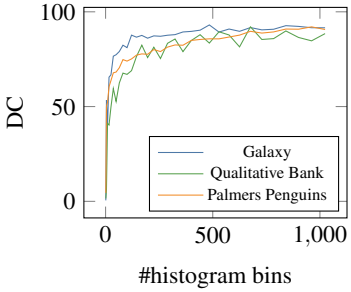

Figure 4: *DC for a varying number of histogram bins.*

use the unique z-Score (see Section 3.3) as an initial guess for the Nelder-Mead algorithm. Since the initial guess and Nelder-Mead both are deterministic, our optimization always results in the same scaling. Thus, our method fulfills Definition 1.

Parameters. The Nelder-Mead algorithm uses few parameters for which standard choices apply. The DSC and CDM quality measures are free of any further parameters. The DSR measure yielded good results with the number of bins being $10,000$, with 100 intervalls per axis. The DC measure requires a histogram bin size or the number of bins, respectively (Section 4.3). It is certainly an option to find the optimal number of bins when evaluating DC, however, this is rather costly – and empirically, without significant benefit. In our experiments, histograms with 100 bins showed good results.

We visualize the effect of varying the DC parameter in a parameter study: Figure 4 varies the histogram bin size. It becomes evident that the result of the DC significantly depends on the number of bins as this also influences the average number of points per bin. When increasing the number of bins, DC increases and approaches its maximum value 100 in the limit. This indicates that the number of bins shouldn't be too large.

Runtime improvement. Our method is generally efficient and applicable for a large number of points p, given a decent performant projection like PCA and performant quality measures like DC and DSR. Other projections like t-SNE and quality measures like CDM become a bottleneck. The worst case for the CDM is an approximately equal ratio of points per cluster, which results in a runtime of $O(p^2)$. Therefore, its direct usage is not an option for large data. We easily remedy this by approximating the resulting scaling from a random subset of points per cluster. In this case, we prescribe a constant total number of points and draw random samples per class in a way that the ratios of class sizes are preserved. This results in a runtime cost that depends only on the number of classes rather than on the number of points. Our experiments show that a total number of points in the order of 1 000 already yields good results (see Section 8).

7. Results

Scaling invariant projections. We applied our method to several datasets. Table 1 gives an overview of the runtimes, Table 2 of the quality measures. More datasets are visualized in the additional matrials. Figures 6 to 9 show scatter plots of four datasets for different projections (in rows) and scalings (columns): The first row use the PCA, the second t-SNE with the $perplexity = 50$ and the third

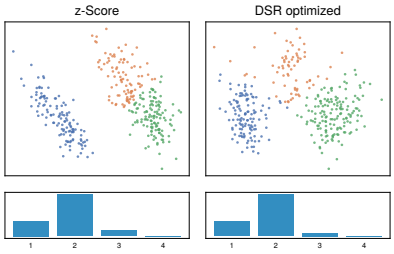

Figure 5: *The* Penguins *dataset projected via a PCA onto 2D with class labels obtained from a K-Means clustering with $k = 3$. The data was z-scaled and optimized via our method using the unsupervised DSR metric. The corresponding scaligns are shown below.*

metric MDS. On each column – from left to right – the original unscaled data, standardized data (Z-score), and our scaling invariant projections with optimized DSC, CDM and DC and DSR. The three selected projections are common scaling invariant ones and are thus a good fit for our application. The scatter plots are color-coded to distinguish different labels.

The *Palmers Penguins* dataset [HHG20], visualized in Figure 6, is a high quality alternative to the popular *Iris* dataset. It measures different attributes of three penguin species from three islands in the Palmer Archipelago, Antarctica, with the species being the classification target. Utilizing the PCA, the three classes form clusters that are coherent in the unscaled plot. However, they overlap severely. One cluster in the standardized version is clearly separated, the other two however are still overlapping. Our DSC, DC and DSR versions solve this issue by separating these two clusters more clearly.

Figure 5 displays the z-Score and DSR optimized scaled PCA of the *Penguins* dataset where the class labels were obtained via the popular K-Means cluster algorithm and $k = 3$. Applying the adjusted rand index (ARI) score, which measures the similarity between two data clusterings by correcting for the chance grouping of elements, providing a value between -0.5 (no agreement) and 1 (perfect agreement), for each clustering and the ground truth yields a value of 0.729 for the z-Scored scaling and 0.850 for the DSR optimized one. Our optimized scaling thus yields a significant better result over the common practice, without even utilizing the underlying class labels.

Comparing the z-Scored scaling with our DSR optimized one for the *Penguins* dataset in Figure 5 yields an interpretable result: it tells us that the first two dimensions, culmen length and culmen depth in *mm*, should be emphasized and the last two, flipper length in *mm* and mass in *g*, should have less influence to optimize the visual structures of the dataset.

The *Galaxy* dataset [Cle93] visualized in Figure 7, comprises 323 observations of radial velocity from a spiral galaxy. The measurements contain information about the east-west and north-south coordinates of these points, as well as angles in degrees and radial position. The target value represents the velocity w.r.t. being above or below a threshold (1593.62 km/s). For this dataset, the DSC optimized scalings separate both clusters more clearly than the Z-score scaled one in all projections. Furthermore, a new star-shaped structure is revealed (in the PCA as well as the MDS version), where

Table 1: *Datasets used in our experiments with their runtimes.*

Dataset	#instances	#dimensions	#classes	Projection	DSC [s]	CDM [s]	DC [s]	DSR [s]
Galaxy [Cle93]	323	4	2	PCA	1.004	0.095	0.051	0.027
				TSNE	49.001	49.489	51.027	54.600
				MDS	8.510	24.817	46.370	24.248
Palmers Penguins [HHG20]	342	4	3	PCA	0.004	0.063	0.034	0.010
				TSNE	73.240	56.130	53.999	60.109
				MDS	13.900	24.052	26.160	29.450
Qualitative Bankruptcy [KH03]	250	6	2	PCA	0.004	0.041	0.003	0.002
				TSNE	35.481	26.322	28.034	32.690
				MDS	11.743	15.509	4.661	4.271
Covertype (sampled to $p = 1185$) [BD99]	581 012	54	7	PCA	1247.570	7.963	1327.69	1.318
				TSNE	2714.324	3195.874	2105.342	1003.827
				MDS	1608.229	739.392	1160.887	1389.492

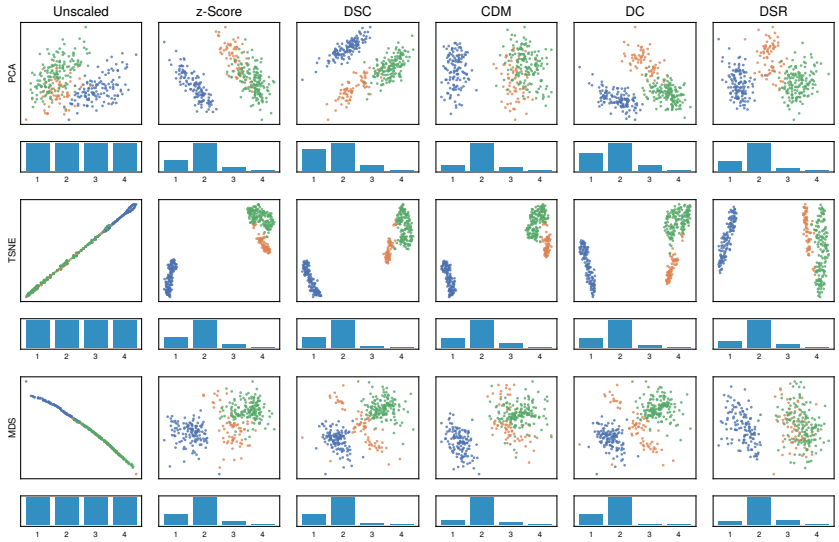

Figure 6: *Scatter plots of three different projections to 2D (rows) with six different initial scalings (columns) of the* Penguins *dataset. The corresponding scaling of the initial data is displayed beneth them. The first two columns are common best practices for choosing a scaling (none and Z-score), the other four are optimized for the given measure.*

the data points seem to be located on lines through the origin. The CDM separates both clusters almost linearly in the PCA and MDS version. The other scalings do not seem to improve or worsen the visualization. The DSR MDS one seem to suggest that the linear structure present in the other scalings like the unscaled one is not intrinsic to all dimensions.

The *Qualitative Bankruptcy* dataset [KH03] in Figure 8 expresses the bankruptcy status based on expert knowledge with 6 categorical features, which we mapped to integers. PCA with the trivial scalings in the first two columns cannot separate the two classes. In contrast, our approach finds three more optimal scalings (for DSC, CDM and DC) that provide a better distinction between the two classes. The CDM t-SNE version suggests that the fifth dimension should be emphasized, since the visual structures seem to better separate both clusters. The DSR and DC MDS version both seem to better distinct both clusters emphasizing the third, fourth and fifth dimension.

Figure 9 shows the *Covertype* dataset [BD99], which classifies forest cover types from cartographic variables. The *Covertype* dataset itself is considerably larger than the other examples in this work: There are 581 012 observations with 54 different features (dimensions). The size of this dataset limits the performance of our proposed algorithm. We address this by using a random subsampling for the optimization (see Section 6), which yields good results for the full dataset but in less than five minutes for the PCA and consistently under one hour for the other projections. The seven classes overlap significantly for the trivial scalings, whereas the visual separation is improved by our method for the PCA DC and DSR version. This is also reflected in the scores, the DC measure scored 65 for the z-Scored and 71 for the DC optimized one. The DSR t-SNE version

Table 2: *Values of the Quality Measures.*

Dataset	Projection	DSC (lower is better) Initial	DSC Optimized	CDM (higher is better) Initial	CDM Optimized	DC (higher is better) Initial	DC Optimized	DSR (higher is better) Initial	DSR Optimized
Galaxy [Cle93]	PCA	0.567	0.548	23620	25510	63.4	70.5	0.027	0.030
	TSNE	0.560	0.526	27157	36108	57.5	66.8	0.028	0.032
	MDS	0.563	0.526	22640	24367	66.0	68.7	0.028	0.031
Palmers Penguins [HHG20]	PCA	0.362	0.029	15598	21027	74.0	95.0	0.033	0.034
	TSNE	0.348	0.020	22732	31656	51.7	98.6	0.015	0.033
	MDS	0.363	0.053	16164	19384	51.7	95.7	0.014	0.034
Qualitative Bankruptcy [KH03]	PCA	0.02	0.0	9589	14387	98.9	100	0.010	0.010
	TSNE	0.004	0.01	11558	16835	98.2	100	0.012	0.013
	MDS	0.02	0.0	10114	12683	97.6	100	0.015	0.023
Covertype (sampled to $p = 1185$) [BD99]	PCA	0.730	0.552	131610	210738	50.0	66.3	0.090	0.106
	TSNE	0.900	0.614	214930	225922	56.0	66.3	0.115	0.118
	MDS	0.790	0.513	143073	143194	50.3	61.9	0.098	0.102

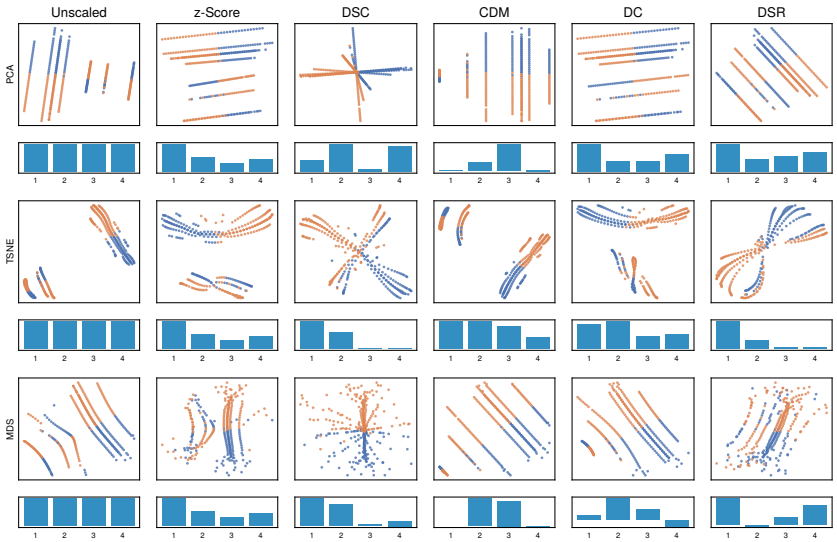

Figure 7: *Scatter plots of three different projections to 2D (rows) with six different initial scalings (columns) of the* Galaxy *dataset. The corresponding scaling of the initial data is displayed beneth them. The first two columns are common best practices for choosing a scaling (none and Z-score), the other four are optimized for the given measure.*

suggest another possile view on the data, avoiding the formation of small blobs. The MDS one does not significantly improve. Since t-SNE as well as metric MDS can not be efficiently computed for the size of this dataset only the subset of 1185 points are plotted.

Performance. All examples were processed using a machine equipped with an 8C/16T AMD Ryzen 7 5700G @ 4.673GHz 32 GB RAM. We implemented our method in Julia and user particularly the `Optim.jl` [MR18] package, which provides the numerical optimization algorithm (Section 6). For the optimization, we set a tolerance of 10^{-8} indicating convergence and a limit of 100 iterations. Table 1 summarizes timings.

8. Discussion and limitations

Our results show that, given labeled input data with arbitrarily scaled dimensions, our method computes unique and deterministic – i.e., scaling invariant – projections. The visualized scatter plots show that our projections often provide a better view and communicate more information than using simple ad-hoc scaling, e.g., by data normalization. Three of the four measures were explicitly proposed and recommended for the automatic analysis of visualizations [SA15], i.e., to visually reveal features like clusters in 2D plots. Therefore, our method computes scaling invariant projections that should be "optimal" for visualization: the goal is to provide an expressive visualization as a single one-shot image. Our results show that our method seems to reach this goal for the majority of example datasets. However, for some datasets, there is no or no good

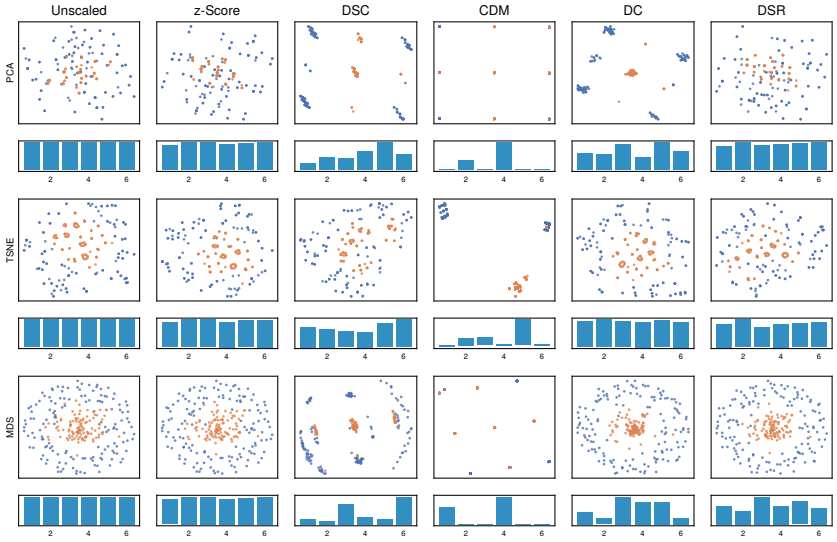

Figure 8: *Scatter plots of three different projections to 2D (rows) with six different initial scalings (columns) of the* Qualitative Bankruptcy *dataset. The corresponding scaling of the initial data is displayed beneth them. The first two columns are common best practices for choosing a scaling (none and Z-score), the other four are optimized for the given measure.*

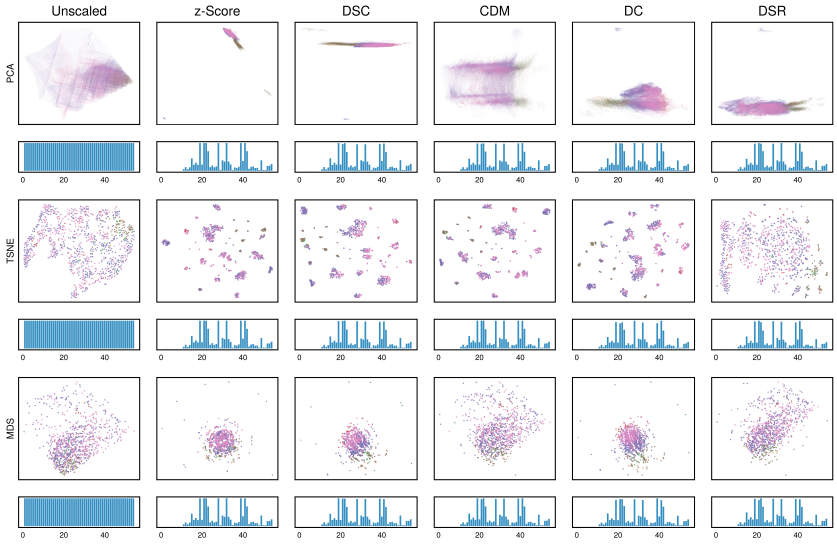

Figure 9: *Scatter plots of three different projections to 2D (rows) with six different initial scalings (columns) of the* Covertype *dataset. PCA plots all points, t-SNE and MDS the random subset of 1185 points. The corresponding scaling of the initial data is displayed beneth them. The first two columns are common best practices for choosing a scaling (none and Z-score), the other four are optimized for the given measure.*

expressive separation of classes through 2D projection possible (see the MDS projection of the *Covertype* dataset in Figure 9). This is certainly a limitation of our method, which is probably inherent to any method that uses projections to 2D. Finally, we think that using our method is always beneficial: First, there is a fair chance that our optimal scaling gives significantly better visualization than trivial preprocessing like normalization at reasonable computational costs (see the PCA projection of the *Penguins* dataset in Figure 6). Second, our method doesn't give degenerate results, in particular, the visualizations are always en par with trivial methods. Our method can also only find structures that are intrinsic to the dataset, projection and scaling, we can not generate new structures (i.e., false positives). Note that we do not make a claim that the method does not introduce additional distortions. In fact, if the chosen quality measure does not contain a part dealing with distortions, our algorithm may minimize this at the cost of having more distortions than

e.g. the original (unscaled or normalized) data set. We consider this a desired behavior if the dimensions are unrelated, because the original data and its scaling is an arbitrary choice. If the quality criterion considers distortions, our approach will result in lower (or equal) distortions than the original projection.

For the majority of our experiments, using DSC and DC as quality measures resulted in similar visualizations. For *Covertype*, the use of CDM and DC yields significantly different results: the visualization using DC shows less structure in the shapes of clusters. But also for few lower-dimensional datasets, the visual quality depends on the quality measure. Consider, e.g., *Qualitative Bankruptcy* dataset: Maximizing DC for the PCA yields a perfect score of 100 as all buckets contain only one class of points, i.e., the histogram bins separate clusters perfectly. In comparison, although the DSC score is not perfect, the result seems slightly better: classes are effectively separated, and there is less spread than for DC. In this case, DC seems limited as reducing the number of histogram bins does not improve the results.

The generated scalings do not change singnificantly for some datasets (e.g. the *Covertype* dataset), the resulting output of the projections however do. This implies a certain sensitivity for each projection and might be studied further in the future.

Our method is bound by the performance of the projection rather than the quality measure. This can be remedied by random subsampling (see Section 6). The CDM and DSR results for the PCA shown in Figure 9 were generated this way in under 10 seconds with only 0.2 % (i.e., 1185 points) of the data used and all 54 dimensions. There was hardly any visual difference compared processing the full dataset, for which the optimization took more than a day.

In some sense, the interactive iPCA [JZF*09] is similar to our method: instead of defining an objective function and automatically finding optimum by numerical methods, the user varies the scaling of dimensions interactively and is provided with instant visual feedback from the PCA. This means, the user takes the perspective of the multivariate optimization algorithm, where the objective function is "replaced" by visual perception, and the steps towards an "optimum" (i.e.varying scale) is purely guided by intuition and/or trial and error. Certainly, it is very unlikely that a user performs similarly to a numerical multivariate optimization method. Even assuming perfect intuition, there is hardly any chance that the search space can be sampled sufficiently even if there are only few dimensions. However, finding an optimum is not the goal of iPCA: instead of one single visualization, this method focuses on the process of interactively and visually exploring the search space, i.e., the visual transitions that are implied by scale variation. In this sense, our method could be used to provide a start configuration for iPCA or possibly multiple start configurations from varying quality measures and their parameters.

Our approach certainly has limitations. We already mentioned that not all datasets can be well-separated using only the scaling as degree of freedom. There is another important limitation that is inherent to the visual quality measures. First, DSC, CDM and DC expect classified input data, i.e., there exists a class label for each data point such that the optimal scaling is computed in a *supervised* manner. This is a significant restriction on the input to our method. Second, there are datasets, for which some or all quality measures are not meaningful or there may be better quality measures.

Here is an example: In Section 2 we introduced the *doublesine* dataset (see Figure 1, right). The 2D data points are not labeled. Depending on the scaling of the axes, there is either no cluster or one single 1-manifold cluster that appears either as a horizontal or as a vertical sine wave. So all points are in one single class, and we cannot apply CDM or DC. At the same time, there is no unique ground truth for scaling as we cannot discriminate between the horizontal and the vertical wave structure. This simple benchmark dataset is challenging as we need to construct an appropriate quality measure, and we need to deal with ambiguity. We leave this as an open problem for future work.

9. Conclusions

We presented an approach to general scaling invariant projections that is based on numerically maximizing the visual quality of a 2D projection. For this, we build upon and adapt existing quality measures that are maximized by varying the scaling of dimensions. The result is an optimal scaling and the resulting projection. We demonstrate the utility of our method for various supervised datasets, and show that first, we effectively achieve scaling invariance, second the resulting projections show good visual separation of classes, and third the method is efficient enough to process large datasets.

Our method is a first step towards scaling invariant projections for data visualization. We see significant potential for future work:

A promising future research is to study the interdependence of dimensions: In this work, we assume that all dimensions of the input data are independent. This is often not true, but instead, dimensions may be partially dependent such that scaling one dimension implies scaling others. An example is a location tag with x and y coordinates: We would expect that it makes no sense to scale or change the unit of only one coordinate.

Finally, the scaling itself can be generalized: Instead of linear scaling, i.e., by a constant factor per dimension, one could think of selecting other scaling functions such as an exponential map or a logarithm. One could compare the effect to visualizations using, e.g., logarithmic axes such as a semi-log plot or log-log plot, to cope with exponential behavior of the input data.

Acknowledgements

This work was partially supported by DFG grant TH 692/21-1.

Supplemental Materials

A file with four more visualizations of datasets is submitted in the supplemental materials. They were omitted due to the space restrictions.

Except for the *doublesine* dataset (Figure 1, right), all datasets considered in this paper are from publicly available data repositories or papers. The creation of the *doublesine* dataset to explain scaling dependence of 2D scatter plots is a contribution of this paper. The *doublesine* dataset is part of the additional materials and will be published with the paper.

References

[BD99] BLACKARD J., DEAN D.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture 24* (1999), 131–151. doi:10.1016/S0168-1699(99)00046-0. 7, 8

[BNHL14] BRADEL L., NORTH C., HOUSE L., LEMAN S.: Multi-model semantic interaction for text analytics. In *IEEE VAST* (Oct 2014), pp. 163–172. doi:10.1109/VAST.2014.7042492. 3

[BSL*08] BUJA A., SWAYNE D. F., LITTMAN M. L., DEAN N., HOFMANN H., CHEN L.: Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics 17*, 2 (2008), 444–472. doi:10.1198/106186008X318440. 3

[CBP09] CHOO J., BOHN S., PARK H.: Two-stage framework for visualization of clustered high dimensional data. pp. 67 – 74. doi:10.1109/VAST.2009.5332629. 3

[CJK82] COX L. H., JOHNSON M. M., KAFADAR K.: Exposition of statistical graphics technology. In *Proceedings of the Statistical Computing Section, American Statistical Association* (1982), pp. 55–56. 10

[Cle93] CLEVELAND W. S.: *Visualizing Data*. Hobart Press, 1993. 6, 7, 8

[CLKP10] CHOO J., LEE H., KIHM J., PARK H.: ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE VAST* (2010), pp. 27–34. URL: http://dblp.uni-trier.de/db/conf/ieeevast/ieeevast2010.html#ChooLKP10. 3

[DKM06] DWYER T., KOREN Y., MARRIOTT K.: Ipsep-cola: An incremental procedure for separation constraint layout of graphs. *IEEE TVCG 12* (2006), 821–8. doi:10.1109/TVCG.2006.156. 3

[DLH80] DE LEEUW J., HEISER W.: Multidimensional scaling with restrictions on the configuration. *Multivariate Analysis 5* (1980). 4

[Dri12] DRIEGER P.: Visual Text Analytics using Semantic Networks and Interactive 3D Visualization. In *EuroVA 2012: International Workshop on Visual Analytics* (2012), Matkovic K., Santucci G., (Eds.), The Eurographics Association. doi:10.2312/PE/EuroVAST/EuroVA12/043-047. 3

[EHM*11] ENDERT A., HAN C., MAITI D., HOUSE L., LEMAN S., NORTH C.: Observation-level interaction with statistical models for visual analytics. pp. 121 – 130. doi:10.1109/VAST.2011.6102449. 3

[FHSW13] FINK M., HAUNERT J., SPOERHASE J., WOLFF A.: Selecting the aspect ratio of a scatter plot based on its delaunay triangulation. *IEEE TVCG 19*, 12 (2013), 2326–2335. 4

[Fis36] FISHER R. A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 7 (1936), 179–188. 3

[Flu97] FLURY B.: *A First Course in Multivariate Statistics*. Springer Texts in Statistics. Springer, 1997. URL: https://books.google.de/books?id=3e4ei8f90GcC. 3

[GH12] GAO F., HAN L.: Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications 51* (2012), 259–277. doi:10.1007/s10589-010-9329-3. 5

[GRM10] GARG S., RAMAKRISHNAN I. V., MUELLER K.: A visual analytics approach to model learning. In *IEEE VAST* (Oct 2010), pp. 67–74. doi:10.1109/VAST.2010.5652484. 3

[GSS*20] GÖRTLER J., SPINNER T., STREEB D., WEISKOPF D., DEUSSEN O.: Uncertainty-aware principal component analysis. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 822–831. doi:10.1109/TVCG.2019.2934812. 3

[HHG20] HORST A. M., HILL A. P., GORMAN K. B.: allisonhorst/palmerpenguins: v0.1.0, July 2020. URL: https://doi.org/10.5281/zenodo.3960218, doi:10.5281/zenodo.3960218. 6, 7, 8

[HL14] HAN F., LIU H.: High dimensional semiparametric scale-invariant principal component analysis, 2014. arXiv:1402.4507. 4

[HLS13] HOSMER D. W., LEMESHOW S., STURDIVANT R. X.: *Applied Logistic Regression*. Wiley, 2013. URL: http://dx.doi.org/10.1002/9781118548387, doi:10.1002/9781118548387. 10

[Hot33] HOTELLING H.: Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology 24* (1933), 417–441. 1, 3

[HR03] HINTON G. E., ROWEIS S. T.: Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, Becker S., Thrun S., Obermayer K., (Eds.). MIT Press, 2003, pp. 857–864. URL: http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf. 3

[HS07] HEINE C., SCHEUERMANN G.: Manual Clustering Refinement using Interaction with Blobs. In *Eurographics/ IEEE-VGTC Symposium on Visualization* (2007), Museth K., Moeller T., Ynnerman A., (Eds.), The Eurographics Association. doi:10.2312/VisSym/EuroVis07/059-066. 3

[JCC*11] JOIA P., COIMBRA D., CUMINATO J. A., PAULOVICH F. V., NONATO L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. doi:10.1109/TVCG.2011.220. 3

[JFSK15] JÄCKLE D., FISCHER F., SCHRECK T., KEIM D.: Temporal mds plots for analysis of multivariate data. *IEEE TVCG 22* (2015), 1–1. doi:10.1109/TVCG.2015.2467553. 3

[JZF*09] JEONG D. H., ZIEMKIEWICZ C., FISHER B., RIBARSKY W., CHANG R.: Ipca: An interactive system for pca-based visual analytics. *Comput. Graph. Forum 28* (2009), 767–774. doi:10.1111/j.1467-8659.2009.01475.x. 3, 10

[KCJ97] KILPATRICK D., CAMERON-JONES M.: Numeric prediction using instance-based learning with encoding length selection. In *International Conference on Neural Information Processing* (1997). URL: https://api.semanticscholar.org/CorpusID:64473593. 10

[KH03] KIM M.-J., HAN I.: The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Systems with Applications 25*, 4 (2003), 637–646. URL: https://www.sciencedirect.com/science/article/pii/S0957417403001027, doi:https://doi.org/10.1016/S0957-4174(03)00102-7. 7, 8

[KKE10] KEIM E. D., KOHLHAMMER J., ELLIS G.: Mastering the information age: Solving problems with visual analytics, eurographics association, 2010. 3

[Kru64] KRUSKAL J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (Mar 1964), 1–27. URL: https://doi.org/10.1007/BF02289565, doi:10.1007/BF02289565. 3, 4

[Llo82] LLOYD S. P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28* (1982), 129–137. 4

[LT18] LEHMANN D. J., THEISEL H.: The lloydrelaxer: An approach to minimize scaling effects for multivariate projections. *IEEE TVCG 24*, 8 (2018), 2424–2439. 4

[LWBP14] LIU S., WANG B., BREMER P.-T., PASCUCCI V.: Distortion-Guided Structure-Driven Interactive Exploration of High-Dimensional Data. *Computer Graphics Forum* (2014). doi:10.1111/cgf.12366. 3

[Mah36] MAHALANOBIS P. C.: On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta) 2* (1936), 49–55. 4

[MH18] MCINNES L., HEALY J.: Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv e-prints 1802.03426* (2018). 3

[ML14] MOLCHANOV V., LINSEN L.: Interactive Design of Multidimensional Data Projection Layout. In *EuroVis - Short Papers* (2014), Elmqvist N., Hlawitschka M., Kennedy J., (Eds.), The Eurographics Association. doi:10.2312/eurovisshort.20141152. 3

[MR18] MOGENSEN P. K., RISETH A. N.: Optim: A mathematical optimization package for Julia. *Journal of Open Source Software 3*, 24 (2018), 615. doi:10.21105/joss.00615. 5, 8

[MSH09] M. SIPS B. NEUBERT J. L., HANRAHAN P.: Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum* (2009). URL: http://graphics.uni-konstanz.de/publikationen/Sips2009SelectingGoodViews, doi:10.1111/j.1467-8659.2009.01467.x. 4

[NA18] NONATO L., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE TVCG* (2018). doi:10.1109/TVCG.2018.2846735. 3

[NM65] NELDER J. A., MEAD R.: A Simplex Method for Function Minimization. *The Computer Journal 7*, 4 (1965), 308–313. URL: https://doi.org/10.1093/comjnl/7.4.308, arXiv:https://academic.oup.com/comjnl/article-pdf/7/4/308/1013182/7-4-308.pdf, doi:10.1093/comjnl/7.4.308. 5

[OKMM15] OSIPYAN H., KRULIŠ M., MARCHAND-MAILLET S.: A Survey of CUDA-based Multidimensional Scaling on GPU Architecture. In *2015 Imperial College Computing Student Workshop (ICCSW 2015)* (Dagstuhl, Germany, 2015), Schulz C., Liew D., (Eds.), vol. 49 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 37–45. URL: http://drops.dagstuhl.de/opus/volltexte/2015/5479, doi:10.4230/OASIcs.ICCSW.2015.37. 3

[PNML08] PAULOVICH F., NONATO L., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE TVCG 14* (2008), 564–75. doi:10.1109/TVCG.2007.70443. 3

[PSN10] PAULOVICH F. V., SILVA C. T., NONATO L. G.: Two-phase mapping for projecting massive data sets. *IEEE TVCG 16*, 6 (2010), 1281–1290. 3

[RL15] RIECK B., LEITTE H.: Persistent Homology for the Evaluation of Dimensionality Reduction Schemes. *Computer Graphics Forum* (2015). doi:10.1111/cgf.12655. 3

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE 290* (2000), 2323–2326. 3

[SA15] SEDLMAIR M., AUPETIT M.: Data-driven evaluation of visual quality measures. *Computer Graphics Forum 34*, 3 (June 2015), 201–210. URL: http://dx.doi.org/10.1111/cgf.12632, doi:10.1111/cgf.12632. 3, 4, 8

[Sam69] SAMMON J. W.: A nonlinear mapping for data structure analysis. *IEEE Trans. Comput. 18*, 5 (1969), 401–409. URL: https://doi.org/10.1109/T-C.1969.222678, doi:10.1109/T-C.1969.222678. 3

[SBVLK09] SCHRECK T., BERNARD J., VON LANDESBERGER T., KOHLHAMMER J.: Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization 8*, 1 (2009), 14–29. URL: http://dx.doi.org/10.1057/ivs.2008.29, doi:10.1057/ivs.2008.29. 3

[SC88] SIEGEL S., CASTELLAN N.: *Nonparametric statistics for the behavioral sciences*, second ed. McGraw–Hill, Inc., 1988. 3

[SSS*14] SACHA D., STOFFEL A., STOFFEL F., KWON B. C., ELLIS G., KEIM D. A.: Knowledge generation model for visual analytics. *IEEE TVCG 20*, 12 (Dec 2014), 1604–1613. doi:10.1109/TVCG.2014.2346481. 3

[SSZ*16] SACHA D., SEDLMAIR M., ZHANG L., LEE J. A., WEISKOPF D., NORTH S. C., KEIM D. A.: Human-centered machine learning through interactive visualization: review and open challenges. In *ESANN* (2016). 3

[SZS*17] SACHA D., ZHANG L., SEDLMAIR M., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE TVCG 23*, 1 (Jan 2017), 241–250. doi:10.1109/TVCG.2016.2598495. 3

[TAE*09] TATU A., ALBUQUERQUE G., EISEMANN M., SCHNEIDEWIND J., THEISEL H., MAGNORK M., KEIM D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. pp. 59 – 66. doi:10.1109/VAST.2009.5332628. 4

[TdSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323. URL: https://www.science.org/doi/abs/10.1126/science.290.5500.2319, arXiv:https://www.science.org/doi/pdf/10.1126/science.290.5500.2319, doi:10.1126/science.290.5500.2319. 3

[TGH11] TALBOT J., GERTH J., HANRAHAN P.: Arc length-based aspect ratio selection. *IEEE TVCG* (2011). URL: http://vis.stanford.edu/papers/arclength-banking. 4

[Tor52] TORGERSON W. S.: Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (Dec 1952), 401–419. URL: https://doi.org/10.1007/BF02288916, doi:10.1007/BF02288916. 1, 3

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html. 3

[VDMPVdH09] VAN DER MAATEN L., POSTMA E., VAN DEN HERIK J.: Dimensionality reduction: a comparative review. *J Mach Learn Res 10* (2009), 66–71. 3

[VPN*10] VENNA J., PELTONEN J., NYBO K., AIDOS H., KASKI S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res. 11* (2010), 451–490. URL: http://dl.acm.org/citation.cfm?id=1756006.1756019. 3

[WL10] WISMÜLLER A., LEE J. A.: Recent advances in nonlinear dimensionality reduction, manifold and topological learning, 2010. 3

[WS06] WEINBERGER K. Q., SAUL L. K.: An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2* (2006), AAAI'06, AAAI Press, pp. 1683–1686. URL: http://dl.acm.org/citation.cfm?id=1597348.1597471. 3