applied sciences

Article

Getting over High-Dimensionality: How Multidimensional Projection Methods Can Assist Data Science

Evandro S. Ortigossa [1,†], Fábio Felix Dias [1,†] and Diego Carvalho do Nascimento [2,*,†]

[1] Institute of Mathematics and Computer Science, University of São Paulo, São Carlos 13566590, Brazil; evortigosa@usp.br (E.S.O.); f_diasfabio@usp.br (F.F.D.)
[2] Departamento de Matemática, Facultad de Ingeniería, Universidad de Atacama, Copiapó 1530000, Chile
* Correspondence: diego.nascimento@uda.cl
† These authors contributed equally to this work.

Citation: Ortigossa, E.S.; Dias, F.F.; Nascimento, D.C.d. Getting over High-Dimensionality: How Multidimensional Projection Methods Can Assist Data Science. *Appl. Sci.* **2022**, *12*, 6799. https://doi.org/10.3390/app12136799

Academic Editor: José Salvador Sánchez Garreta

Received: 26 May 2022
Accepted: 27 June 2022
Published: 5 July 2022

Abstract: The exploration and analysis of multidimensional data can be pretty complex tasks, requiring sophisticated tools able to transform large amounts of data bearing multiple parameters into helpful information. Multidimensional projection techniques figure as powerful tools for transforming multidimensional data into visual information according to similarity features. Integrating this class of methods into a framework devoted to data sciences can contribute to generating more expressive means of visual analytics. Although the Principal Component Analysis (PCA) is a well-known method in this context, it is not the only one, and, sometimes, its abilities and limitations are not adequately discussed or taken into consideration by users. Therefore, knowing in-depth multidimensional projection techniques, their strengths, and the possible distortions they can create is of significant importance for researchers developing knowledge-discovery systems. This research presents a comprehensive overview of current state-of-the-art multidimensional projection techniques and shows example codes in Python and R languages, all available on the internet. The survey segment discusses the different types of techniques applied to multidimensional projection tasks from their background, application processes, capabilities, and limitations, opening the internal processes of the methods and demystifying their concepts. We also illustrate two problems, from a genetic experiment (supervised) and text mining (non-supervised), presenting solutions through multidimensional projection application. Finally, we brought elements that reverberate the competitiveness of multidimensional projection techniques towards high-dimension data visualization, commonly needed in data sciences solutions.

Keywords: high-dimensional data; dimensionality reduction; multidimensional scaling; artificial intelligence; information visualization

1. Introduction

Datasets generated nowadays have become increasingly larger and more structurally complex, generated from a diverse range of sources, such as relational and NoSQL databases, web pages, texts, images, recordings, video, and others. The analytical effort to explore these datasets is a challenging problem, although they could load valuable content for discovering and understanding phenomena in many knowledge domains. The more data collected, the more complex the analysis, thus causing graphical perception to be infeasible due to the amount of information displayed on a screen [1]. Multidimensional data are those with $m \geq 4$ attributes for each data object and may be represented as families of curves on the m-dimensional space [2]. Line and bar charts have long represented data in several applications. Nevertheless, line-based visual metaphors are not scalable for treating multiple variables at the same time, thus leading to problems of visual cluttering and occlusions [3].

An efficient and comprehensive representation of multidimensional data are beyond the capabilities of simple line and bar charts due to the complexity of the multidimensional structured data and the inability of these simple techniques to deal with large data volumes on a screen. Information Visualization (InfoVis) aims to develop and apply visual representations to model and understand attribute values, relationships, and extraction of information from data [4,5]. In this context, a class of InfoVis techniques has been a stand out for some time: the multidimensional projections. These techniques generate injective mappings that transform multidimensional spaces into visual spaces (2D or 3D), preserving structures and similarities of the original spaces as much as possible, such as relationships between data instances or clusters' presence [6–8]. This transformation is accomplished through mathematical operations to embed the number of attributes to 2 or 3, allowing m-dimensional data objects to be represented in Cartesian space.

Although it is not a recent concept, the research into new multidimensional projection techniques development has been intensified in recent years. This rise of interest is due to the wide range of applications that can benefit from visual representations of large datasets with a large number of attributes [9], and the research into projection methods has promoted the creation of visual tools that reveal relevant patterns and trends hidden in multidimensional data. Examples of successful applications involving projection techniques can be found in the more diversified domains [6,10–12]. The current literature on multidimensional projections has followed the recent developments, with many works reviewing fundamental concepts, proposing taxonomies, and performance evaluations [7,13–18].

However, due to the profusion of new developments, we have identified some things missing in the literature. Moreover, the different terminologies adopted in previous papers and reviews can be somewhat confusing for the researchers starting into the subject and even for those with some previous knowledge but (maybe) not specialized in InfoVis methodologies. For example, each technique often cannot handle all the data types, as each technique is tailored to efficiently handle one (or some) specific data type(s). In addition, all the projection techniques require careful data pre-processing to explore their embedding abilities properly. Nevertheless, we believe that multidimensional projection techniques are highly informative tools that can add analytical capabilities to data scientists within the context of knowledge discovery and data mining tasks; thus, a comprehensive discussion regarding simplifying the projection technique environment is a need.

In this sense, we aim to present to the reader that multidimensional data analyses go far from simple PCA. We will do it with a sufficiently detailed discussion of the leading and more advanced different types of techniques applied to multidimensional projection tasks through a vocabulary that unifies the many and different terms and definitions found in previous works. We will also address taxonomies and essential concepts and considerations that must be taken into account by the reader's interest in applying projection techniques on multidimensional data, from the appropriate data treatment to the possible distortions and limitations of the obtained results.

The manuscript is structured as follows: Section 2 presents the research methodology adopted for the collection of papers; Section 3 addresses the ground theory to understand the multidimensional projection domain; Section 4 investigates multidimensional projection approaches and their applications; Section 5 shows discussions and applications; Section 6 examines the challenges and future research directions concerning multidimensional projections; finally, the Section 7 presents our final comments.

2. Method of the Systematic Review

A content analysis of the published literature was conducted to understand better how the multidimensional projection techniques have evolved over the last few years. Such an analysis systematically evaluates the available forms of communication, identifying and classifying critical contributions to the field, clarifying trends and practices, and indicating future and open research possibilities. Therefore, we elaborated so that the research goals could be achieved, and we established objective criteria to define the relevant literature and the appropriate reporting of the findings.

Literature Search Procedure

To conduct a comprehensive search into multidimensional scaling multidimensional projection literature, we collected and combined the research material mainly from three databases: Association for Computing Machinery (ACM) Digital Library (https://csur.acm.org/, accessed on 12 February 2022), IEEE Xplore Digital Library (https://ieeexplore.ieee.org, accessed on 10 April 2022), and Elsevier's Scopus (http://scopus.com, accessed on 10 April 2022), and search engines, such as Google Scholar (https://scholar.google.com, accessed on 10 April 2022), Elsevier's ScienceDirect (https://sciencedirect.com/, accessed on 20 March 2022), and Thomson Reuters's Web of Science (http://apps.webofknowledge.com, accessed on 10 April 2022), were used in association with the databases.

We query "multidimensional projection", "dimensionality reduction", and "multidimensional scaling" terms, mainly restricted to (but not only) the 2005–2022 period and related to publications' title, abstract, and keywords. Such a period was chosen to cover most of the published literature not reported by previous surveys and including seminal surveys. The following two criteria were employed over the search results to select publications for a further revision:

- Papers published in peer-reviewed journals as articles and available online in English are the priority sources. In addition to these papers, we extended our scope to conference proceedings, arXiv e-Prints, thesis, dissertations, and books;
- Papers explicitly employ multidimensional data and multidimensional projection techniques. Then, we excluded those articles that only list multidimensional projection in keywords, allude to multidimensional data as datasets, or apply multidimensional projection without further explanation or reference to the specific methodology employed to processing and presenting the information.

Papers that did not fulfill at least one of the selection criteria were removed from the review. The queries returned 183 papers; after second filtering, only 131 were read in full. Duplicates were removed; the first selection criteria were applied, and the second criteria were used after carefully reviewing the texts. Finally, we reported 123 references here.

3. Ground Theory

This section provides an overview of definitions that should be considered when applying and developing multidimensional projection techniques. Our intent is not only to provide a comprehensive survey of multidimensional projection methods but also to discuss essential features and particularities in the context of data analysis and visualization.

3.1. Data Multidimensionality

Formally, a multidimensional dataset can be expressed as a set of n instances $X = \{x_1, x_2, \ldots, x_n\}$, in which each instance contains a vector of m items, $x_i = \{v_1, v_2, \ldots, v_m\}$. Each x_i vector can be seen as a data object and is represented by a subset of attributes of different data types, such as integers, reals, binaries, text, categorical, images, and others. When x_i has more than three attributes, the dataset cannot be represented in simple visualizations such as those based on the Cartesian plane. In this case, the vector produces families of curves in the m-dimensional space [19].

3.2. Basic Terminology

The literature regarding multidimensional projection, or even multidimensional data, often utilizes many different terms to name similar concepts, which can lead to some mistakes. For example, it presents terms such as "multidimensional", "multivariate", or even "multivalued" for defining datasets in which each data object is composed of a set of attributes. However, these terms are not consistently used [20], and we will adopt the term "multidimensional" to denote datasets composed of multiple attributes with or without dependence on each other.

Since the beginning of this text, we have named each vector x_i as data objects, but there are terms such as "instance", "observation", "data item" or just "item", "record", "array" or even "point". We named the vectors x_i from a dataset X as "data objects" or "instances".

Another terminology is related to x_i attributes, and one can find several terms denoting them, such as "variable", "dimension", "property", "characteristic", or "feature". We used throughout the text the terms "attribute" or "feature".

In addition, we will use the term "point" exclusively to name instances transformed into the visual space. We finish this subsection by noting that matrix notation from mathematics is standard to denote the x_i data objects such as vectors. However, considering the context of data analysis, we argue that the terms adopted here are in line with precise data and mathematical terminology. This kind of explanation is due to the multidimensional projection domain being closely related to matrix theory on mathematical grounds.

3.3. Multidimensional Projection

A multidimensional dataset X, with n instances $x_i \in \mathbb{R}^m$, can be represented as a matrix $X_{n \times m}$, where each row is a vector x_i, and the columns represent the vector attributes. According to Ware [21], high-dimensional datasets are those with $m \geq 12$ attributes. Since the visual space is limited to three dimensions, when the m-dimensionality increases, the complexity of representing and interpreting data also increases.

In this context, techniques devoted to dimensionality reduction aim to present multidimensional data from high-dimensional spaces as points in dimensionality-reduced spaces. Thus, it is possible to use fewer dimensions to describe the original data, compacting large datasets and reducing the computational effort required in their processing. According to Pudil and Novovicova [22], there are two main categories of dimensionality reduction methods: feature selection and feature extraction.

Feature selection identifies the non-significant features related to the analysis task, omitting them to generate a subset containing only useful attributes that can efficiently describe the input data [23]. Based on the assumption that the original dataset may have redundant (or even irrelevant) attributes, the removal process of non-significant features will generate a subset of useful features that automatically result in a reduced data space [24]. In several cases, feature selection is sufficient to carry out data analysis. However, there are situations in which it is challenging to identify a significant attributes subset, limiting the use of feature selection to the pre-processing data phase. As feature selection often does not result in a dataset of two or three dimensions, these methods are not part of this research. (for a comprehensive overview of feature selection methods, see Chandrashekar and Sahin [23]).

On the other hand, feature extraction methods try to reduce data dimensionality by mapping the original dataset X from an m-dimensional space to a new dataset Y of p dimensions, with $p < m$. The mapping process uses a single function or a group of transformation functions to map the data, retaining the essential characteristics of the original dataset [25]. When $p \in \{1, 2, 3\}$, there is a particular class of multidimensional data mapping techniques called Multidimensional Projections (MDP) [7].

MDP maps multidimensional data to the Cartesian visual space, seeking to preserve some information from the original space in the mapped (projected) space, such as the distance relationships between the data instances [6]. Consequently, it is possible to create a graphical representation from the projected points, to take advantage of the human perceptual abilities to recognize structures or visual patterns based on similarities, particularly those related to groups of elements [9]. Projection techniques have been attracting the attention of the Information Visualization community for some time due to their broad applicability as an analytical tool. Making the mapped data convenient for visualization, tasks that involve distance exploration or neighborhood relations can be streamlined [6].

Tejada et al. [26] mathematically defined the multidimensional projection concept as

Definition 1. *Let X be a set of n data objects in $\mathbb{R}^m$, with $m > 3$ and $\delta : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ a distance measure between the instances in $\mathbb{R}^m$; Y be a set of n points in $\mathbb{R}^p$, with $p \in \{1,2,3\}$ and $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ a distance measure between points in $\mathbb{R}^p$. A multidimensional projection technique can be described as a function $f : X \to Y$ that aims to make $|\delta(x_i, x_j) - d(f(x_i), f(x_j))|$ as close as possible to zero, $\forall\, x_i, x_j \in X$.*

Similarity and dissimilarity represent somewhat vague concepts, as they do not yet have a well-established definition in the literature and may present different interpretations depending on the application domain [7]. In the context of MDP, and to prevent any inaccuracy, we will adopt a geometric definition of (dis)similarity. Therefore, we established that dissimilarity is a numerical measure used to indicate the "proximity" (distance) between two data objects. The higher the dissimilarity value, the more distant and distinct the objects are, according to some criterion or comparison function [27]. Inversely, a similarity measure indicates how similar two objects are, with high values of similarity denoting close and similar objects.

In this context, the concept of distance has a central role in multidimensional data comparison and projection. An interesting set of dissimilarity metrics is the Minkowski distances calculated as:

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^k \right)^{\frac{1}{k}}, \quad k = 1, \ldots, \infty, \tag{1}$$

being k a modification parameter. The changing of k generates well-known functions, such as Manhattan distance ($k = 1$), Euclidean distance or L_2 norm ($k = 2$), and Maximum distance or L_{max} norm ($k = \infty$). Several MDP techniques utilize Euclidean distance as the dissimilarity metric, such as the ones proposed by Maaten and Hinton [28], Joia et al. [12], and McInnes et al. [29]. Nevertheless, depending on the nature of the data and the application domain under analysis, Euclidean distance may not be the best way to express similarities or differences between data objects. A measure is considered a metric if it satisfies the metric space postulates [30], and some measures do not satisfy them. Therefore, it may be convenient to apply measures that do not meet the metric space properties in some cases. A deep analysis of the different types of existing measures is beyond the scope of this paper.

Calculation of (dis)similarities can generate distortions owing to (i) significant differences between vectors Euclidean norms, and (ii) features with values much larger or much smaller than others can lead to projection results and force it to ignore the importance of other features [31]. According to Joia et al. [12], distortions are inevitable in the multidimensional projection process, but it is possible to keep them as small as possible.

To avoid these situations, a scaling process is applied to each vector (vector normalization) or data feature (features normalization or standardization). Scaling is one of the more important pre-processing steps of projection techniques because it can unify the feature space by forcing the unitary Euclidean norm of vectors or forcing features to satisfy a specific statistical property [32].

Figure 1 illustrates the application of normalization before checking the similarities between two data vectors. In this example, two sequences, A and B, have different graphical curves in terms of range and magnitude (horizontal and vertical shifting). However, the distance between A and B is proportional to the shifting, which can hide information about the sequences' similarity. In other words, if a proper normalization is not applied before the projection task, the projection method may overestimate the distances, which leads to distorted information regarding the correct amount of (dis)similarities among the data vectors (indicated by the shaded area between the curves in Figure 1, right side). After a vector normalization, one can determine the true similarities between the distinct curve profiles [33].

Following this example, each item x_i of a multidimensional dataset X can be normalized with:

$$\hat{x}_{ij} = \frac{x_{ij}}{\|x_i\|}, \tag{2}$$

being j less than m (the number of data features), and $\|\cdot\|$ the Euclidean norm. The importance and effects of normalization were extensively documented in papers such as Keogh and Kasetty [33].

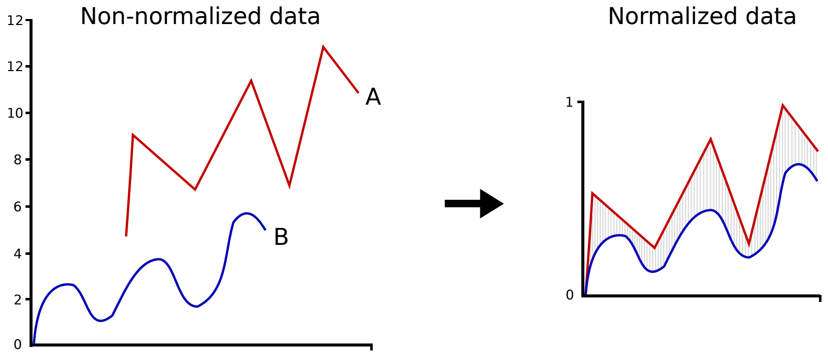

Figure 1. Vector normalization is applied to improve the (dis)similarity measure process. The shaded area between A and B sequences represents the real difference in magnitude between them. Source: elaborated by the authors.

Another example of transformation is the z-score standardization applied to each j-column of an X dataset, making columns' mean equal to zero and standard deviation equal to one. The process consists in subtracting all values of a j-column of its mean and dividing the result by the standard column deviation as in

$$\hat{x}_{ij} = \frac{(x_{ij} - \bar{x}_j)}{\sigma_j}. \tag{3}$$

And another common transformation is the *min-max* normalization, which converts the j-column range $[min_{x_j}, max_{x_j}]$ to a new range $[min_{new}, max_{new}]$ with a simple linear transformation:

$$\hat{x}_{ij} = (x_{ij} - min_{x_j}) \times \left(\frac{max_{new} - min_{new}}{max_{x_j} - min_{x_j}} \right) + min_{new}. \tag{4}$$

the new range is frequently $[0, 1]$, and variations with other ranges and application of logarithm and square root functions can be performed [32].

This section presented elementary approaches focused on multidimensional data pre-processing real numerical values. However, other transformations may be more appropriate depending on the data domain (categorical, textual, images, and others) and the problem to be solved. There are several other ways to transform the data and measure its dissimilarities according to the data context, and a detailed review on the subject can be found in Zezula et al. [30].

3.4. Taxonomies Used in the Multidimensional Projection Area

The application of multidimensional projection techniques results in groups of points embedded in a p-dimensional space. We define a group, or a class, as a set of instances; however, there are some conceptual differences. A group can be formed by points perceptually close in the visual space, generated by a clustering algorithm that applies some dissimilarity measure in the data space. On the other hand, a class comprises objects that share some intrinsic meaning previously known or revealed by a classification algorithm that describes or distinguishes the objects belonging to the class [7].

Different approaches have been proposed to map multidimensional data into visual space, and the literature classifies them with some terminologies that share common characteristics. Among the taxonomies found in the literature, we can cite the best known and usually adopted as follows:

(I) According to the transformation type

Linear techniques are based on transformations that create linear combinations of the data attributes, mapping them into a new space with a reduced dimensionality [9]. Formally, linear methods map the data using a function $f : X \rightarrow Y$, which satisfies the condition $f(\alpha x_i + \beta x_j) = \alpha f(x_i) + \beta f(x_j), \forall\, x_i, x_j \in X$ and $\alpha, \beta \in \mathbb{R}$ [25]. Although many MDPs apply linear resources in their processes, few approaches are truly linear, according to the strict mathematical definition [7].

Linear methods are often computationally efficient and relatively simple to develop. Once the transformation is calculated, the mapping of each instance in the Cartesian space is performed with matrix multiplication. Nevertheless, they cannot properly handle complex structures (such as data with nonlinear relationships between its attributes), achieving unsatisfactory results in many real scenarios [34]. This characteristic tends to generate representations with considerable distortions, which is a severe issue for visualization tasks.

Nonlinearity arises in many problems, such as physical phenomena modeling, and sometimes functions cannot satisfy linearity conditions that naturally motivate the development of nonlinear projections. Nonlinear techniques aim to minimize an information loss function usually supported by a mechanism able to relate the dissimilarities between the instances in the high dimensional space with the distances between the p-dimensional points [9].

Linear methods can present good results depending on the data distribution, but they rarely outperform nonlinear techniques in terms of the ability to deal with complex structures [16]. Another problem is the addition of new data subsets in the visual space. Linear techniques require complete reprocessing to introduce a perturbation in the dataset because they are based on matrix operations [9]. On the other hand, nonlinear approaches require only a small number of additional iterations to incorporate new data into the projection.

(II) According to the projection nature

The concept of locality is used to distinguish two properties of projections nature: global modeling and local mapping. Global modeling defines methods built in a single transformation [12] and methods that seek to preserve original geometric relationships between all pairs of data instances in the transformed space [6]. That is, global modeling forces close points in the original space to remain close to the projected space, while distant points remain distant in the visual space.

Local mapping techniques are also based on neighborhood information, but they seek to preserve relationships considering only the surroundings of a small neighborhood in the m-dimensional space [6]. Close points to an instance in the high dimensional space must be projected as close as possible to that instance in the visual space. Generally, local methods build a family of transformations to project the data; therefore, instances are not mapped by a single transformation but using a set of local mappings in which each mapping will project a subset of data, preserving the local structures of the dataset.

Local methods can rely on global mechanisms to perform multidimensional projections or combine global and local properties, generating hybrid approaches. For example, a small subset of data samples (often called control points) is defined and projected, relying on global relationships. From this initial mapping, the remaining instances are positioned based only on their neighborhood relationships concerning each control point [12], which maintains the essence of a local mapping.

In summary, global techniques are often accurate in preserving distances average between objects, in global terms, but fail to preserve local relationships, besides being computationally more expensive. Moreover, according to Fadel et al. [6], preserving distances as a whole tends to distort small neighborhoods. Thus, when the purpose is to preserve neighborhoods, local techniques are more suitable than global ones.

(III) Other classifications

The taxonomy presented is the more widely adopted in the literature devoted to MDP. However, other classifications have been proposed to characterize the properties of different strategies. For example, projections can also be classified as interactive and non-interactive according to their ability to admit user interventions during the mapping process. In this sense, weighting mechanisms allow certain instances to have more influence on the mapping, while control points can be used as user-specified "anchors", which is an important step to guide the projection sequence behavior [7]. Sadly, choosing an appropriate set of control points to generate a mapping with less distortion as possible is a complex problem [12].

There are other possible classifications, such as the ones related to method stability, when the introduction of small perturbations in the data does not result in significant changes in the mapping. Another classification addresses multilevel projections that contain resources to allow the organization of large databases through hierarchies, creating different levels of abstraction to reduce the visual clutter [7]. Projection methods can also be classified according to their mathematical formulations related to the decomposition procedures or the optimization methods adopted to perform the mapping. Maaten et al. [16] and Nonato and Aupetit [7] conducted extensive reviews on techniques for reducing dimensionality, classifying them in other subdivisions.

3.5. Evaluation of Projected Spaces

Projection techniques can approximate similar instances and segregate distinct ones, grouping them in the visual space according to some (dis)similarity measure. Nevertheless, as aforementioned, MDPs can generate different levels of distortions in the data neighborhood. Therefore, to assess the distortions and quality of the views created, specialists map real data classes or groups to a color space, allowing them to evaluate the projection results visually. Sadly, this type of assessment can be subjective and complex due to the data cluttering on the screen, and specialists utilize some measures to assess projections' quality.

The best-known measure used to evaluate MDPs quality is the stress function [35], which estimates the amount of information lost during the mapping process of data instances. The function is similar to a standard deviation calculation and is defined as

$$stress = \sqrt{\frac{\sum_{i<j}(d_{ij}-\delta_{ij})^2}{\sum_{i<j}(\delta_{ij})^2}}, \tag{5}$$

where δ_{ij} is the dissimilarity measure in the original space, and d_{ij} is the dissimilarity in the projected space. The stress function can quantify the distortion degree generated through the mapping process concerning the original space.

The results are in the $[0,1]$ range, and the closer to zero, the better the distances preservation, with zero technically indicating a "perfect" projection. Although the stress function helps evaluate the generated mappings, it is common to find divergences between stress and visual projection interpretation [9].

Another measure is the neighborhood preservation index [10], which assesses the rate of neighbors that are close in the high dimensional space and remain close in the projected space. Some projection technique generates neighborhood distortions. For example, two instances close to each other (or similar) in the original space can be projected too far, which is a missing neighbor problem. On the other hand, two instances distant from each other (or dissimilar) in the original space can be projected into the same neighborhood, a problem that is named false neighbor (see Figure 1 in [7]). Note these two kinds of distortions that can affect the neighborhood preservation index. False and missing neighbors impact the projection-based analysis in different ways [36] and can generate a degree of uncertainty during the data analysis task.

The index is computed using:

$$NP_k(i) = \frac{|kNN(y_i) \cap kNN(x_i)|}{|kNN(x_i)|}, \tag{6}$$

where $kNN(\cdot)$ is the set of the k nearest neighbors of some instance x_i in the original or y_i in the projected space [6].

To evaluate the projection of the entire dataset, one can take the mean of $NP_k(i)$ values in the $[0,1]$ range, with results closer to 1 indicating mappings with better preservation of neighborhoods concerning the original space.

The silhouette coefficient [37] is one of the more widely used qualitative measures to check clusters' quality, based on calculating the cohesion and separation between groups of objects. Nonetheless, it has been primarily applied to assess the quality of projected spaces [38,39]. The cohesion a_i of some instance x_i is calculated by averaging distances between x_i with all other instances belonging to the x_i group. The inter-group separation b_i is defined by the minimum distance between x_i and all the other instances belonging to the other groups [12]. Therefore, the silhouette coefficient of some projected data instance x_i is computed as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{7}$$

and the average $Silh$ of all these values determines the average width of the silhouette coefficients of a dataset, considering $Silh \in [-1,1]$, and the higher its value, the better the intra-group cohesion and inter-group separations. That is, projections that present a silhouette close to 1 will have instances of the same group close to each other, with the different groups distant from each other.

The measures described here present features considered an initial step to quantitatively assessing MDPs. However, other measures are developed to evaluate the layout generated by multidimensional projection techniques. A comprehensive analysis of the different qualitative measures applied in the context of projections can be found in Bertini et al. [40].

3.6. Influence of Graphical Perception and Visual Properties in Projection Analysis

Cleveland and McGill [41] defined visual perception as the ability of users to interpret visual encodings and understand the information presented graphically. It is challenging to extract the significance of complex data structures since their instances bear multiple attributes that may vary individually and simultaneously. The well-known *Gestalt* theory states that the global pattern perception of a scene cannot be explained by the sum of parts [42]. In other words, a single element can present a specific context, but when associated with other elements, they can represent different characteristics and contexts.

All in all, users need tools that explore the complex data features to reduce the cognitive load involved in analyzing massive amounts of data, such as multidimensional projection. However, how do we present data to users to explore their cognitive abilities to make the information presented significant? In graphical terms, the *Gestalt* theory formulated some interesting principles to "setup" the scenario more understandable, and we summarized it as follows:

- Proximity principle—visual elements close in the visual space are preattentive (intrinsic and uncontrolled) processed as a group that shares similar features, even if instances are grouped in a not explicit way;
- Similarity principle—elements represented by visual structures that share similar features (size, color, orientation, symmetry, parallelism) are perceptually grouped;

- "Common fate" principle—visual elements that undergo similar visual transformations tend to be mentally grouped. The dynamism of movement helps the viewer to perceive which objects are related to the same action;
- Closure principle—elements delimited in areas with clear contours tend to be visually grouped, even if they are not entirely continuous.

Color, position, size, user interaction, and other graphical properties are essential for a suitable visualization layout. Moreover, a comprehensive MDP tool must present accurate layouts that follow graphical properties that explore the human's cognitive abilities, leading to the visual transformation of raw data into information. The graphic metaphor commonly applied to display data resulting from a projection is the scatter plot [43].

A classical 2D scatter plot maps values of a data object using symbols on the Cartesian plane. For example, let $S \in \mathbb{R}^2$ be a dataset whose s instances record the mass of an element measured over the interval t. Then, for each i point, the i-th symbol position relative to the abscissa axis is horizontally determined through the t_i value in which s_i was measured. At the same time, the position relative to the ordinate axis of that i-th symbol is vertically determined by the s_i value, measured in t_i. The advantage of a scatter plot is the consumption of only a few pixels to represent a single data object, standing out as a space-efficient chart. In addition, they are well-established metaphors commonly used in the scientific and business context [44].

According to McLachlan et al. [45], to learn quickly about visualization tools, one must explore the user experience and intuition; therefore, graphic representations should be familiar to the user. In this sense, scatter plots are straightforward; however, they have limitations and could suffer from visual scalability. Occlusion occurs as the number of points becomes larger than screen resolution [46]; overlapping occurs when there are very close dots on each other, obscuring the individual legibility of the symbols [44]. An important issue regarding the scatter plots in a multidimensional projection context is: that there are no coordinated axes. The reason is that variables belonging to different scalar types are processed to extract similarity information between each instance, and, in the visual space, the transformed data points must express their relative position as information. According to Nonato and Aupetit [7], references such as orthogonal axes perturb the analysis by raising questions about the meaning of the axes, inducing users to read the absolute positions of points in visual space.

In the following sections, we will discuss approaches and techniques that apply the points reviewed in this section.

4. Multidimensional Projection Approaches and Domains

This section addresses the main dimensionality reduction techniques found in the literature, emphasizing the review of well-established multidimensional projection approaches. This methodological choice is justified due to the recent interest in applying this approach in the information visualization context, that is, when the p-dimensional space is 2D or 3D. Here, each method will be described in a subsection containing discussions about its characteristics and properties and some variations and improvements. The following subsections were organized according to the development chronology of the methods.

4.1. Principal Component Analysis

Principal Component Analysis (PCA) was introduced in the earlier 1900s. It was probably the first dimensionality reduction technique and has been well-known and used [47]. PCA maps data from a high dimensional space to a low one by encountering orthogonal linear combinations that better represent original data variability. These combinations are named principal components. The process is performed so that the first component is related to the highest data variance direction; the second component is related to the second higher variance, orthogonal to the previous component, and so on.

One can obtain the components $a_1, a_2, \ldots, a_p$ by calculating the covariance of the m attributes of a dataset X (Section 3.3). Being x_i and x_j two attributes of X and $\bar{x}_i$, and $\bar{x}_j$ their means, the covariance between them is defined as

$$cov(x_i, x_j) = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j) \tag{8}$$

With this formulation, it is possible to build a square matrix $C_{m \times m}$ with each position representing the covariance between each data attribute pair. Following, spectral decomposition can be applied to encounter eigenvectors and eigenvalues by writing C as:

$$C = U \Lambda U^T, \tag{9}$$

such that $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_m)$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, is a diagonal matrix with the eigenvalues of C and $U_{m \times m}$ is the orthogonal matrix with its eigenvectors. Finally, the principal components that represent the p-dimensional space are generated by the application of eigenvectors in the data matrix:

$$A = X \times [u_1, u_2, \ldots, u_p], \tag{10}$$

with each u_i being the columns of the U matrix.

It is possible to obtain the number of principal components equal to the number of original data dimensions. However, the number p of dimensions should be sufficiently large to represent the initial space without significant information loss. It is important to observe that the first components generated by PCA can absorb the more significant part of data variance. This characteristic makes PCA a suitable approach to identifying data trends and patterns [48], beyond weeding out part of data noise and catching data variability in a few dimensions.

Due to numerical and practical questions, PCA is implemented by Singular Value Decomposition (SVD), which can be similarly interpreted as the eigendecomposition aforementioned. The process centers on X columns, subtracting each value by the respective column mean. SVD decomposes the centered matrix with

$$B = USV^T, \tag{11}$$

being U columns left singular vectors, V columns right singular vectors, and S a diagonal matrix with the singular values. Then, the p-dimensional space is created with $A = US$ [49]. In this case, covariance was not calculated, but the centralized data force the meaning of singular vectors and values to be similar to eigendecomposition results.

As a projection technique, PCA preserves more global structures than the local ones, owing to the preservation of the large variances that appear in orthogonal principal axes [7,15]. Nevertheless, it is not capable of well-representing datasets formed by nonlinear relations among attributes [31]. Besides, PCA is unsuitable for representing data with several groups that have distinct variances since, for visualization purposes, just 2 or 3 first components are used [31].

4.2. Multidimensional Scaling

One of the first global multidimensional projection techniques is Multidimensional Scaling (MDS), a classic algorithm that originated from psychophysics, becoming famous after the Torgerson [50] paper. Nowadays, MDS comprises a family of nonlinear methods that seek to define an injective mapping of data objects from a multidimensional space to a lower one, maintaining data distance relations [9]. The way that preserves distances throughout the transformation process determines the differences among approaches based on MDS.

Classical Scaling is the best-known MDS method. This technique builds a transformed space by applying spectral decomposition of a symmetric matrix generated from distances of multidimensional objects [6,27]. More specifically, it seeks to satisfy $\delta(x_i, x_j) = d(y_i, y_j)$, in Euclidean space, with $\delta(x_i, x_j)$ being the dissimilarity between objects x_i and x_j, and $d(y_i, y_j)$ the dissimilarity between projected objects. The step of spectral decomposition is similar to the step demonstrated in PCA (Section 4.1).

The outcomes attained by Classical Scaling are considered satisfactory and precise concerning global distance preservation. However, its computational complexity of $O(n^3)$ bounds the approach application. Thus, applying Classical Scaling in large datasets (more than millions of items) is not viable. To mitigate this problem, some alternatives were developed, such as Landmark MDS (LMDS) [51] and Pivot MDS [52].

Instead of decomposing the complete dissimilarity matrix, LMDS selects an initial subset from the original dataset containing s reference instances named *landmarks*. These reference instances can be selected randomly or by an algorithm that maximizes the distances among them. The approach performs Classical Scaling in the landmarks subset, projecting these reference points on the $\mathbb{R}^p$. Finally, the remaining data instances are mapped on the new space through distance-based triangulation.

LMDS preserves MDS characteristics and is more efficient with its complexity of roughly $O(s^3 + sn)$. Nonetheless, it should be selected at least $p + 1$ landmarks to generate a good projection.

4.3. Sammon's Mapping and Related Projections

In some cases, the obligation of dissimilarities maintenance during the transformation process can be restrictive, generating poor mapping. In this scenario, techniques based on nonlinear optimization emerged. This class of global methods is derived from MDS theory and maps original space to the visual space by minimizing a loss function g in such a way that $d(y_i, y_j) \approx g(\delta(x_i, x_j))$.

Gradient descent is a well-known algorithm for minimization problems, and it is used in the following projection methods. Gradient descent walks in the opposite sense of the function gradient $-\nabla f(\vec{x}(k))$, with steps that have an α size (varying each iteration defined by a line search satisfying the Wolfe conditions or the Barzilai–Borwein method). The step size is defined by applying some algorithm or a heuristic method. Finally, each update step is defined as $\vec{x}(k+1) = \vec{x}(k) - \alpha(k)\nabla f(\vec{x}(k))$ and the algorithm is finished when the gradient function stops the changes or a maximum number of $k > 0$ is reached. The gradient, also known as the slope of a line, is defined as the ratio of the change, and whenever the gradient is perpendicular to $\vec{x}(k)$, then it will be the negative reciprocal of the original line.

Kruskal [35] was the first one to model the projection problem as an optimization problem that seeks to generate a layout that minimizes the quadratic difference between original data dissimilarities and the projected data distances. This quadratic difference is well-known as the *stress* function (Equation (5)) and will be identified here by S_T. Then, Kruskal [35] used Gradient Descent to find values that minimize the *stress* function. With $\vec{x}(k)$ being a vector with all the n instances of X after the k-th interaction, the minimization process updates the mapping with the following equation:

$$\vec{x}(k+1) = \vec{x}(k) - \alpha\left(\frac{\partial S_T(k)}{\partial \vec{x}(k)} \Bigg/ \left|\frac{\partial S_T(k)}{\partial \vec{x}(k)}\right|\right) \tag{12}$$

In this process, the normalization of the gradient is applied not to change the magnitude of $\vec{x}(k)$.

Kruskal's method is related to Sammon's Mapping [53], one of the best-known dimensionality reduction techniques in the InfoVis area. The latter introduces a loss function normalization in a different way, which can be figured out when comparing Equation (5) (used by Kruskal) and Equation (13) (applied by Sammon). Sammon [53] mapping also uses Gradient Descent to minimize the information lost in the transformation process while preserving the global data dissimilarities of the original space by finding a local minimum solution [9,54].

$$E = \frac{1}{\sum\limits_{i<j} \delta(x_i, x_j)} \sum_{i<j} \frac{(\delta(x_i, x_j) - d(y_i, y_j))^2}{\delta(x_i, x_j)} \tag{13}$$

Note that Equation (13) is weighted by the sum of $\delta(x_i, x_j)^{-1}$, which gives more importance to small dissimilarities. This characteristic enables Sammon's Mapping to project data with high dimensionality; however, it can generate distortions in multidimensional data with nonlinear relations, an issue that also appears in Kruskal's method [9].

Sammon [53] defined the update steps as the following equation:

$$y_i(k+1) = y_i(k) - \beta\Delta_i(k), \tag{14}$$

such that the step size β can be into the interval $0.3 \leq \beta \leq 0.4$ and defined by:

$$\Delta_{it}(k) = \frac{\partial E(k)}{\partial y_{it}(k)} \Bigg/ \left| \frac{\partial^2 E(k)}{\partial y_{it}^2(k)} \right|. \tag{15}$$

Another issue related to Sammon's and Kruskal's methods is the quadratic computational complexity ($O(kn^2)$ operations, see Table 1). To mitigate it, Pekalska et al. [54] developed a strategy that, in the first step, maps a subset with s samples to visual space using Sammon's Mapping. Then, the remaining samples are positioned by applying some interpolation method, such as triangulation, neural network, or linear transformation. With this change, the approach attained a good precision and improved the efficiency with the complexity of $O(s^3 + sn)$ [12]. On the other hand, the new approach needs representative samples of "high quantity" to perform the initial projection, with the authors suggesting half of the dataset, $n/2$.

Table 1. Multidimensional projection methods. The column "Complexity" presents the computational load related with the amount of instances (n), dimensions (m), iterations (k), samples (s), and graph edges (E).

Technique	Transformation	Nature	Complexity
PCA	Linear	Global	$O(m^3)$
Classical MDS	Linear	Global	$O(n^3)$
Kruskal	Nonlinear	Global	$O(kn^2)$
Sammon's	Nonlinear	Global	$O(kn^2)$
FastMap	Nonlinear	Global	$O(n)$
Chalmers	Nonlinear	Local	$O(n^2)$
Pekalska	Nonlinear	Global	$O(s^3 + sn)$
Isomap	Nonlinear	Global	$O(n^3)$
Chalmers Hybrid	Nonlinear	Local	$O(n\sqrt{n})$
L-Isomap	Nonlinear	Global	$O(sn \log n)$
SNE	Nonlinear	Local	$O(n^2)$
Force Scheme	Nonlinear	Global	$O(n^2)$
LMDS	Nonlinear	Global	$O(s^3 + sn)$

Table 1. *Cont.*

Technique	Transformation	Nature	Complexity
LSP	Nonlinear	Global	$O(n^3)$
HiPP	Nonlinear	Global	$O(n\sqrt{n})$
t-SNE	Nonlinear	Local	$O(n^2)$
Glimmer	Nonlinear	Global	$O(n^2)$
PLMP	Partially linear	Global	$O(n^3)$
PLP	Nonlinear	Local	$O(n\sqrt{n})$
LAMP	Nonlinear	Local-Hybrid	$O(sn)$
LoCH	Nonlinear	Local	$O(n\sqrt{n})$
UMAP	Nonlinear	Local	$O(n^{1.14})$
TopoMap	Nonlinear	Global	$O(n \log n)$
GRMP	Nonlinear	Local	$O(n + \|E\|)$

Source: elaborated by the authors.

4.4. Isometric Feature Mapping

Another important variation of Classical Scaling is the Isometric Feature Mapping (Isomap) [55], a global nonlinear projection technique. Isomap transforms the distance relationships among multidimensional data instances before projecting them into the visual space [31]. Isomap aims to capture the data's topological (manifold) structure by computing geodesic distances by turning the dataset into a graph structure and considering distances based on the shortest path between vertices. Isomap is probably the first projection technique to resort to topological data analysis mechanisms to achieve dimensionality reduction [56].

Isomap builds an undirected weighted graph $G = (V, E)$, such that $v_i \in V$ represents a data instance x_i and $w_{ij} = \delta(x_i, x_j)$ is the weight function. The graph edges can be created with the *k-nearest neighbor* (k-NN) approach, which considers a data distance to define each $e_i \in E$. Therefore, the k closest vertices to a specific vertex v are connected. With this process, the δ function information is surrogated by the shortest path between points calculated by some path algorithm, such as Dijkstra [57]. Moreover, the graph's adjacent matrix D is considered as the "distance" matrix of data, with cells related to data with no edge filled up by ∞ value.

Finally, the visual space is created by applying some projection technique, such as MDS, in the D matrix. Sadly, the calculation of distance and the spectral decomposition affect computational performance that attains complexity $O(n^3)$. The topological stability of Isomap depends on the correct choice of k [58]. Moreover, isometric mapping of high dimensionality into low one is possible only in particular conditions, resulting in mistakes and distortions [7].

Silva and Tenenbaum [59] created a more computationally efficient version of Isomap, named L-Isomap. First, it selects s landmarks to generate the graph representing relationships among the landmarks and the other data instances. After that, the LMDS technique is applied to project the landmarks and approximate the positions of the remaining points. It reduces the Isomap complexity to $O(sn \log n)$.

4.5. FastMap

Faloutsos and Lin [60] created the FastMap projection, a global nonlinear technique that reaches high efficacy but generates low accuracy, depending on the structure of the projected data. FastMap maps objects from an $\mathbb{R}^m$ space into an $\mathbb{R}^p$, with $p < m$, where each data item is projected into p orthogonal hyperplanes representing projected data coordinates. The approach's core is to project items in a particular line segment contained on a hyperplane.

The process starts with a set of instances $\mathcal{O}$ and the distance matrix $\mathcal{D}$ generated with the Euclidean distance function δ of the instances. As depicted in Figure 2, two instances, O_a and O_b, furthest apart, are selected from the high-dimensional space. This selection requires $O(n^2)$ computations; however, Faloutsos and Lin [60] also proposed a linear heuristic to perform it. Firstly, it selects an arbitrary instance O_r. In the sequel, it chooses the farthest apart instance from O_r to be the O_a pivot. Finally, it selects the instance that is farthest apart from O_a to be the O_b pivot. The line O_aO_b is into a hyperplane Y, and it will represent the first coordinate of the new projected space $\mathbb{R}^p$.

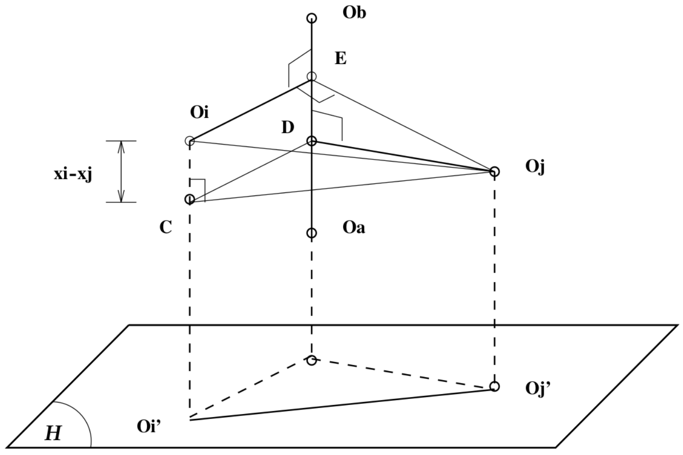

Figure 2. Projection of O_i and O_j into H hyperplane that is orthogonal to the previous one. The points O_a and O_b remain in the previous hyperplane and the triangles $O_aO_iO_b$ and $O_aO_jO_b$, such as the respective points E and D, were considered to generate previous coordinates. Source: Faloutsos and Lin [60].

Then, each object O_i is projected into O_aO_b line through the calculation of x_i coordinate that manipulates the cosine-law formulation as in Equation (16). In Figure 2, x_i is calculated using the $O_aO_iO_b$ triangle and x_j is calculated using $O_aO_jO_b$.

$$x_i = \frac{\delta(O_a, O_i)^2 + \delta(O_a, O_b)^2 - \delta(O_b, O_i)^2}{2\delta(O_a, O_b)} \tag{16}$$

The next step involves calculating distances in the hyperplane Y, generating a new $\mathcal{D}$ matrix. As described by Equation (17), the new distance is related to the previous one and the difference between the coordinates of the new point-object:

$$\delta(x_i', x_j')' = \sqrt{\delta(O_i, O_j)^2 - (x_i - x_j)^2}, \quad i, j = 1, \ldots, n \tag{17}$$

Lastly, other dimensions can be generated by applying the same process. For example, according to Figure 2, starting now with the new distance matrix, encountering new pivot objects in a hyperplane H, calculating the position of all objects on the pivot line, and creating another distance matrix. This process guarantees that the current line O_aO_b, and consequently the hyperplane H that contains it, is orthogonal to the previous one. The process continues until all p dimensions are calculated.

FastMap is based on classical Euclidean geometry, requiring only data dissimilarities and employing just two representative samples to guarantee orthogonality of dimensions to lead data projection [6]. The algorithm has linear complexity, being faster than other ones. Sadly, it presents high levels of information loss and can not represent nonlinear datasets properly.

4.6. Force-Based Placement

Force-based techniques are simple approaches applied to data projection that consider data instances connected by "virtual springs" and attempt to minimize the sum of the forces that act upon each data instance. Low dimensional positions are randomly initialized, and the repulsion forces between the springs adjust positions towards balancing [9]. The forces are calculated by $\delta(x_i, x_j) - d(y_i, y_j)$, being δ and d dissimilarity functions in original and projected space, respectively. Therefore, distant points in the original space repel each other in the projected space, and close points in the original space attract each other in the projected space.

Sadly, the mapping accuracy is impaired by the computational complexity of $O(n^3)$ (n iterations of $O(n^2)$ force calculations), making its application in large datasets unfeasible. Chalmers [61] handled this problem with a limited-size neighborhood of each data instance. After randomly initializing the projected space, considering an object x_i, the process creates a V_i subset that contains k-near-neighbor of x_i and calculates δ_{maxi} with the maximum dissimilarity between x_i and the members of V_i. In an iterative process, a subset S_i is loaded until reaching k' instances. Each iteration randomly selects a new x_j object that does not belong to V_i. If $\delta(x_i, x_j) < \delta_{maxi}$, then x_j is inserted into V_i (since the corresponding point is deleted from V_i) and δ_{maxi} is updated; otherwise, x_j is inserted into S_i. When S_i is fulled with k' instances, Equation (18) returns the resultant force to be applied to y_i that is linearly proportional to $\delta(x_i, x_j) - d(y_i, y_j)$.

$$F_i = \sum_{v \in V_i} F_{iv} + \sum_{s \in S_i} F_{is} \tag{18}$$

Therefore, instead of calculating $n(n-1)$ forces in each iteration, Chalmers [61] seeks to preserve original distances in projected space, considering a reduced subset of instances. Despite the computational complexity improvement and preservation of small neighborhoods with linear iterations [6], $O(n^2)$ operations to balance the layout still hinder its application in real applications [9].

Morrison et al. [62] also tried to reduce the computational complexity with a hybrid version of the technique based on a sampling strategy. The process generates a random sample with $\sqrt{(}n)$ instances and projects them with Chalmers' model. The $n - \sqrt{(}n)$ remaining data instances are interpolated with a modified version of the Brodbeck and Girardin [63] strategy. These modifications reduced complexity to $O(n\sqrt{n})$, but it can be inaccurate when applied to complex data structures or can get stuck in a local minimum [64].

Another approach was proposed by Tejada et al. [26] and named *Force Scheme*. This projection moves all points in a N_i neighborhood towards an y_i point instead of moving y_i towards its neighbors. In other words, each y_i attracts or repulses other y_j points during an iteration. Positions are updated with a force fraction considering the residual distance function between original and projected spaces, defined as

$$\Delta = \frac{\delta(x_i, x_j) - \delta_{min}}{\delta_{max} - \delta_{min}} - d(y_i, y_j), \tag{19}$$

where δ_{max} and δ_{min} are the more significant and the smallest dissimilarities in the original space, respectively. The formulation respects original global dissimilarities, but this characteristic allows the large dissimilarities to dominate the projection process and distort small local neighborhoods. Consequently, in datasets that contain instances with high dissimilarities, the process tends to concentrate points in the out-most edges of the projection. Force Scheme can generate accurate layouts and reach stabilization with few iterations compared with other force projections. Sadly, the amount of operations in each iteration is $O(n^2)$, which is high computational complexity.

Ingram et al. [64] developed *Glimmer*, a technique that seeks to reduce the processing time of forces calculation using the processing power of GPUs (Graphics Processing Units). The algorithm projects the data with a multi-level variation of Chalmers' model, starting with one data instance and adding new instances until the whole dataset is mapped to the projected space. The method presents a good data cohesion and separation performance, reaching small *stress* values. Sadly, even with the efforts to reduce computational complexity, force-directed projections are unfeasible for huge datasets [12].

4.7. Stochastic Neighbor Embedding

Hinton and Roweis [65] developed the Stochastic Neighbor Embedding (SNE), a nonlinear statistical technique based on the minimization of the Kullback–Leibler (KL) divergence [66] that takes probabilities distributions to represent dissimilarities from original and projected spaces. SNE models two Gaussian distributions, one P over the original m-dimensional space and another Q over the projected p-dimensional space (initialized with random positions). In this scenario, pairs of instances with small Euclidean distances are associated with high probability values, and pairs with large distances have low probabilities.

Equation (20) defines the conditional probability $p_{i|j}$ between each par of instances x_i and x_j, being δ the Euclidean distance and σ_i related to the size of the x_i neighborhood, being smaller in dense regions and bigger in the sparse ones [67]. The best value of σ_i is encountered by a binary search in the $[1, \mathit{perplexity}]$ range in such a way that the Shannon entropy of p probabilities reaches $\log_2 \mathit{perplexity}$, being $\mathit{perplexity}$ frequently defined by the user in the $[5, 50]$ range [65]. Meanwhile, Equation (21) defines probabilities in p-dimensional space, being $\sigma_i = 1/\sqrt{2}$ without generality loss. In both spaces, self-probabilities are null, i.e., $p_{i|i} = q_{i|i} = 0$.

$$p_{i|j} = \frac{\exp(-\delta(x_i, x_j)^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\delta(x_i, x_k)^2 / 2\sigma_i^2)} \tag{20}$$

$$q_{i|j} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} \tag{21}$$

KL divergence between probability distributions P_i and Q_i is minimized by the application of the gradient descent algorithm [68] to discover the best values of p-dimensional coordinates [67] such as in Equation (22).

$$\arg\min_{y} \sum_i \mathrm{KL}(P_i \| Q_i) = \sum_i \sum_j p_{i|j} \log \frac{p_{i|j}}{q_{i|j}} \tag{22}$$

This optimization process can be interpreted as a system with attraction/repulsion forces between y_i and all y_j points because it approximates similar points and segregates the dissimilar ones.

Sadly, KL divergence generates space distortions due to its asymmetric characteristics and is sensitive to the crowding problem [28]. These issues were mitigated by Maaten and Hinton [28] with the introduction of *t-Distributed Stochastic Neighbor Embedding* (t-SNE), which applies to the original space a symmetrized version of conditional probabilities, such as

$$p_{ij} = p_{ji} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{23}$$

being $p_{ii} = 0$. In the projected space, the authors replaced the Gaussian distribution with the t-Student distribution because the latter is simpler and faster to compute while maintaining similarities with the former distribution [28]. In addition, at the projected space is defined $q_{ij} = q_{ji}$ and $q_{ii} = 0$.

Both SNE and t-SNE can represent global structures and reveal local data characteristics, preserving some levels of data neighborhood. The t-SNE and other similar techniques are robust against norm concentration, a typical characteristic of the curse of dimensionality [7], because of the shift-invariant similarities [69]. Although attaining good results if compared to other methods, in t-SNE projections, the observed distance between clusters is unreliable and sensitive to parametrization (perplexity parameter) [70], which can generate a misleading effect of cluster distance and shape resulting from the local nature of optimization and how hyperparameters are set up [8]. Nonetheless, t-SNE presents computational complexity equal to $O(n^2)$, which impairs practical applications, but Maaten [67] developed an accelerated version that reaches $O(n \log n)$ through approximations.

4.8. Least Square Projection

The Least Square Projection (LSP) [9] is a nonlinear global projection that preserves neighbor relationships at a high level and allows user interactions. It consists of two steps: (i) a small subset of data samples is carefully chosen and projected with some MDS approach; (ii) the positions of the remaining data instances are calculated with a linear system that considers the previous projected points and a Laplacian matrix operator that maps the neighbor relations of each instance to place it close to its nearest points.

The initial subset of samples, named here as control points, leads the geometry of the final map, and its position in the projected space significantly impacts the LSP result; thus, they must accurately represent the different groupings of instances from the original space. To select them, a dataset X with n instances is split into $s = \sqrt{n}$ clusters using the *k-medoids* algorithm [13], and the clusters' medoids are then considered as the control points. According to Paulovich et al. [9], the definition of s with the squared root of dataset size can accurately represent the whole dataset, keeping LSP complexity feasible. An MDS technique or a user interaction-based method is applied to lay the control points out, allowing the user to manipulate the mapping process.

In the next step, LSP maps the remaining instances x_i with a strategy that generates a list V_i with k neighbors of x_i. The coordinates of the projected point y_i are generated through the solution of Equation (24) that bounds y_i points to their neighbor convex hull. Each α_{ij} weights the impact of neighbors over y_i position and when $\alpha_{ij} = \frac{1}{|V_i|}$, y_i turns into V_i centroid.

$$y_i - \sum_{x_j \in V_i} \alpha_{ij} y_j = 0, \quad \text{, being} \;\; 0 \leq \alpha_{ij} \leq 1 \;\; \text{and} \;\; \sum_j \alpha_{ij} = 1 \tag{24}$$

The solution of Equation (24) reaches the linear system $LY = 0$, where $L_{n \times n}$ is a Laplacian matrix (a matrix representation of a graph) and its elements are defined as:

$$l_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\alpha_{ij} & \text{if } x_j \in V_i, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Sadly, L has no geometric information, and the system solutions can be useless. This issue is handled through the insertion of new lines in L containing geometric information of the projected control points turning the Laplacian system into a new non-homogeneous system $AY = b$, where $A = \left[\frac{L}{S}\right]$, $S_{s \times n}$, and b vector (let us consider $\bar{Y}$ the inner product space) defined as:

$$s_{ij} = \begin{cases} 1 & \text{if } y_j \in \bar{Y}, \\ 0 & \text{otherwise.} \end{cases} \quad \text{e} \quad b_i = \begin{cases} 0 & \text{if } i \leq n, \\ y_i & \text{otherwise.} \end{cases} \tag{26}$$

LSP calculates the system solution with the least square method [68] minimizing $\|AY - b\|^2$ since an exact solution is unlikely. As a result, the final system is sparse and symmetric, which facilitates an interactive solution [9].

LSP presents a good level o neighbor preservation, but there are n variables to solve the system with an $O(n^2)$ complexity, which is infeasible in huge datasets [6]. Then, Paulovich et al. [71] created *Piecewise Laplacian-based Projection* (PLP) to mitigate this complexity issue. PLP splits the dataset and maps the subsets with LSP. The spatial coherence is achieved through the initial projection of the control points, and the computer time is reduced because PLP calculates small systems instead of a big and more complex ones. However, this solution is often less accurate than LSP [6].

4.9. Part-Linear Multidimensional Projection

Part-Linear Multidimensional Projection (PLMP) [72] is a global and partially linear multidimensional projection method that can deal with huge amounts of data due to its low complexity level. It consists of two steps: (i) a nonlinear step that projects a subset of representative samples; (ii) a linear step that projects the remaining instances with a linear transformation, justifying the "part-linear" name.

PLMP selects s control points (see Section 4.8), a subset of representative instances of $\bar{X}$ dataset that are projected with the Force Scheme, generating the $\bar{Y}$ low-dimensional set. In the next step, a linear transformation $\Phi : \mathbb{R}^m \to \mathbb{R}^p$ minimizes the differences between the projected distances and the original dissimilarities, such as

$$\Phi = \underset{\hat{\Phi} \in \mathscr{L}_{m,p}}{\arg\min} \left\{ \frac{1}{\sum_{ij} \delta(x_i, x_j)^2} \sum_{ij} (\delta(x_i, x_j) - \|\hat{\Phi}(x_i) - \hat{\Phi}(x_j)\|)^2 \right\}, \tag{27}$$

$x_i, x_j \in X$ and $\mathscr{L}_{m,p}$ being a space of linear transformations that maps $\mathbb{R}^m$ to $\mathbb{R}^p$.

Sadly, Equation (27) is computationally impractical in large datasets. To mitigate this issue, PLMP determines an approximation $\hat{\Phi}$, which takes information from a subset $\bar{X}$ of the projected control points. Consequently, the final solution is a linear mapping that approximates the transformation of the control points [72].

To execute the approximation, PLMP creates a new system $\hat{\Phi}_{m \times p} \bar{X}_{s \times m} = \bar{Y}_{s \times p}$, where $\bar{X}$ and $\bar{Y}$ are, respectively, the control points and their projections. Each line of $\hat{\Phi}_{m \times p}$ generates a sub-system, and the least square is applied to solve it. The projection of the remaining instances is achieved by repeating the process to Φ (Equation (27)), that is $X\Phi = Y$, a simple matrix multiplication with $O(n)$ computational complexity.

As in the before-mentioned techniques, the users can manually define the position of the projected control points. Nonetheless, PLMP requires a number of control points superior to the data attribute number, that is, $s > m$, to build a good quality layout, and this characteristic can limit its use with very high-dimensional datasets such as text represented with a bag of words.

The results presented by Paulovich et al. [72] highlighted that PLMP is faster than projections such as LSP. Moreover, besides the initial step, PLMP formulation does not need data dissimilarities information, reducing computational complexity.

4.10. Local Affine Multidimensional Projection

Following Section 3.4, projection techniques that apply linear transformations are based on one function that maps high-dimensional data into the visual space. This function can be very generic in dealing with all data characteristics, even when it is optimized to treat each data subset, such as PLMP (see Section 4.9). On the other hand, most of the nonlinear transformations depend on dissimilarity matrices whose calculation demands high computational complexity. Some techniques project representative instances to guide the process to outperform complexity-related issues, but it demands the choice of several samples to execute the initial projection.

In this context, Joia et al. [12] developed the Local Affine Multidimensional Projection (LAMP). This method also uses representative instances (control points) to guide the projection at the first step. It then maps the remaining instances locally through interpolation using a family of orthogonal affine mappings—one linear function for each instance to be projected.

More specifically, LAMP executes a process similar to LSP (see Section 4.8), where s representative samples are randomly selected from a dataset X and projected on the visual space with Force Scheme (see Section 4.6). With this initial information from control points' positions, each remaining instance $x_i \in X$ will be mapped with a linear function $f_X(p) = pM + t$, that matrix M and vector t are unknowns (by taking partial derivatives with respect to t equal to zero, one can write t in terms of M) that minimizes:

$$\sum_{i=1}^{s} \alpha_i \|\bar{x}_i M - \bar{y}_i\|^2, \quad \text{such that} \quad M^T M = I, \tag{28}$$

being $\bar{x}_i = x_i - \frac{\sum_i \alpha_i x_i}{\alpha}$, and $\bar{y}_i = y_i - \frac{\sum_i \alpha_i y_i}{\alpha}$, respectively, a control point and its projection, and $\alpha_i = \frac{1}{\|\bar{x}_i - x\|^2}$ a weight that leads the control points to influence other data instances, where $\alpha = \sum_i \alpha_i$. Finally, the matrix M is going to be computed from the product of the left and right singular vectors derived from the SVD of the vector column $(\bar{x}) \times$ vector line $(\bar{y})$ (for further information, please see Section 3.1 in [12]). Consequently, the more a point has similarities to x_i, the more it affects Equation (28), which reinforces local characteristics of the LAMP process.

Following Joia et al. [12], the orthogonality restriction imposed on Equation (28) guarantees an isometric transformation, avoiding scaling and shear effects. Moreover, the restriction reduced the high error levels generated during the projection of control points and propagated during the other transformations, keeping the (inevitable) distortions as small as possible.

The authors rewrote Equation (28) into matrix notation and generated a *Procrustes Orthogonal problem* [73], a well-known mathematical problem of matrix approximation. This equation can be expressed as $\|AM - B\|_F, \quad \text{tal que} \quad M^T M = I$, being $\|\cdot\|_F$ the Frobenius norm and:

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{x}_1 \\ \sqrt{\alpha_2}\hat{x}_2 \\ \vdots \\ \sqrt{\alpha_s}\hat{x}_s \end{bmatrix}, \quad B = \begin{bmatrix} \sqrt{\alpha_1}\hat{y}_1 \\ \sqrt{\alpha_2}\hat{y}_2 \\ \vdots \\ \sqrt{\alpha_s}\hat{y}_s \end{bmatrix}. \tag{29}$$

Finally, the matrix equation is solved using SVD, being $M = UV$ and $A^T B = UDV$, which demands $O(s)$ operations. As a result, the position of data points is obtained with $y = f_X(x) = (x - \tilde{x})M + \tilde{y}$.

LAMP was classified as a hybrid projection because its behavior can be local or global depending on the number of control points [6], which can be seen as a positive correlation, i.e., the more control points, the more global will be the mapping. Joia et al. [12] performed tests that revealed LAMP generates high-quality layouts even with a small number of control points. The formulation makes LAMP efficient with $O(sn)$ computational complexity.

4.11. Hierarchical Approaches

Hierarchical Point Placement Strategy (HiPP) is a projection technique built to preserve the clustering and segregation of a dataset [10]. It generates a hierarchical structure that allows exploration in several levels of detail. With this asset, HiPP can be applied to larger datasets than several other projections. The authors reported that all process has a computational complexity of $O(n\sqrt{n})$, being n the total number of data instances.

The authors divided the HiPP process into three stages. In the first one, HiPP creates a tree that has all data instances as children of the root node. With *Bisecting k-means* [74], the children's level is split into k clusters. One node is inserted as a child of the ancestor node representing each cluster created. After this, all cluster items are connected with their related group node. This process is recursively applied to each new node until a global threshold of items inside each node is reached. Paulovich and Minghim [10] used a threshold equal to the square root of the dataset instances, $\sqrt{n}$, and a local $k = \sqrt{m}$, with m being the number of items inside the node (group) being divided.

In the second stage, the authors applied LSP to project tree nodes. They represented groups as circles, being its center point related to the group centers, and the size of the circle proportional to the number of items in the group. Thus, the first level is projected with these attributes, placing the circles on the plane. Next, internal data are projected when one interacts with a circle (group), zooming in and out on the structure.

In the third stage, a spreading algorithm removes overlapping in the visual layout. A vector between every two nodes is calculated and used to spread them. The nodes are dislocated in the vector orientation but opposite senses.

Furthermore, the authors used HiPP to explore text datasets. Thus, they present topic terms as labels of each group and colored circles based on topics. The technique also allows split and reassemble groups interactively. HiPP is suitable when one needs to tailor clustering. On the other hand, it does not allow for flexibility of clustering and projection techniques or data types.

To enhance HiPP capabilities, Dias and Minghim [75] proposed the *eXtend Hierarchical Point Placement Strategy* (xHiPP). First and foremost, the authors use three cluster techniques in the tree construction process: *k-means*, *k-medoids*, and a basic *Hierarchical Agglomerative Clustering*. Instead of using $\sqrt{n}$ to decide how many groups will be generated, the authors used Sturges' Rule [76], which defines $k = 1 + 3.3 \times log(n)$. It results in a smaller number of clusters than the previous technique, which is better when one works with a large dataset. In addition, the user also can modify this parameter.

In the second change proposed, the user can choose which projection better fit the data under analysis. It is possible to choose among LSP MDS, Force Scheme, t-SNE, PLMP, LAMP, or the PCA, all previously described in this review. Force Scheme is always used to project internal data to improve performance.

Dias and Minghim [75] allowed xHiPP to cluster data before the projection stage (such as the original approach) and also the projection stage, followed by the clustering stage. It takes advantage of the projection techniques' ability of data points that have come together in projected space and partition the dataset with high precision.

Beyond text exploration tools implemented in the original HiPP, the authors added word clouds to summarize the content of both groups and individual documents. Authors also added functionalities to facilitate the exploration of image datasets, audios, and general data. The groups' medoids are used to represent image sets and spectrograms of audio ones. Heatmap images are employed to map attributes' distribution of general data. The capabilities to present text, image, audio, and value attributes of general data were added to support the examination of these data types. If data are labeled, colors represent the node and predominant group labels.

Following the authors, with these changes and additions, it is possible for the user to combine different methods to find the better set of them that represent the data under analysis.

4.12. Local Convex Hull

The Local Convex Hull (LoCH) [6] is a local technique built to project data embedded into multidimensional sparse space through three steps. In the first step, LoCH generates $\sqrt{n}$ data clusters and finds their medoids (similar to LSP control points seen in Section 4.8), and these medoids are used to calculate distances among clusters. For each x_i data instance, the projection creates a k-neighborhood, considering the data inside the s closest clusters of x_i.

In the second step, the clusters' medoids are projected with Force Scheme [26]. According to Fadel et al. [6] testing, Force Scheme preserves global distance relationships despite the high-precision of its results.

In the final step, the remaining x_i instances are projected, applying an interpolation that considers the previous projected points and the x_i distance to the convex hull of the k-

neighborhood generated in the first step. Based on this idea, the x_i position in the projected space is determined by the linear combination $\hat{y}_i = \sum \alpha_j y_j$; then,

$$\alpha_j = \frac{1}{\delta(x_i, x_j) \cdot \sum\limits_{x_k \in N_i} \frac{1}{\delta(x_i, x_k)}}, \text{being} \begin{cases} N_i & x_i \text{ neighborhood} \\ \alpha_j \geqslant 0 \\ \sum \alpha_j = 1 \\ \delta & \text{distance function in the original space.} \end{cases} \tag{30}$$

With this formulation, the distance relationships are preserved; however, it does not guarantee that $\hat{y}_i$ is placed inside the correct convex hull. Therefore, the initial $\hat{y}_i$ is iteratively moved toward $\tilde{y}_i$:

$$\tilde{y} = \hat{y} + \gamma_i \frac{\overrightarrow{v}}{\| \overrightarrow{v} \|}, \text{being} \begin{cases} \gamma_i = \frac{1}{k_i} \sum\limits_{x_j \in N_i} \left(\tau_j + \sqrt{\delta(x_i, x_j)^2 + \tau_j^2 - \| \overrightarrow{u_j} \|^2} \right) \\ \tau_j = \frac{\overrightarrow{v}}{\| \overrightarrow{v} \|} \cdot \overrightarrow{u_j} \\ \overrightarrow{u_j} \quad \text{a director vector from } \hat{y} \text{ and a point } y_j \text{ in the } x_i \text{ neighborhood.} \end{cases} \tag{31}$$

In this context, γ_i is a preservation factor of the original distances between x_i and its neighborhood N_i. This process needs almost $\sqrt{n}$ iterations to approximate y_i and the convex hull [6]. LoCH is simple to implement and presents a low computational complexity, namely $O(n\sqrt{n})$. Sadly, LoCH characteristics (well suited to multidimensional sparse datasets) limit its application in a large range of applications.

4.13. Uniform Manifold Approximation for Dimension Reduction

Uniform Manifold Approximation and Projection (UMAP) [29] is a multidimensional projection technique with a solid base on geometry and topology theories. UMAP is a manifold learning method that relies on Riemannian geometry and algebraic topology to perform dimensional reduction [77]. The authors highlighted that UMAP is scalable, which turns it suitable in real applications with large datasets. Furthermore, UMAP presents results as good as t-SNE (currently one of the widely used projection techniques) in visualization quality; nonetheless, it better preserves global data structures and is faster than t-SNE. In addition, UMAP was developed with machine learning applications in mind [8].

To perform data mapping, UMAP considers the following assumptions: (i) data are uniformly distributed in a manifold, and (ii) the manifold is locally connected, i.e., there are no isolated points. McInnes et al. [29] used manifold approximations and patching associated with local fuzzy simplicial set [78] to construct a topological representation for original and projected spaces. Following this step, UMAP minimizes the cross-entropy function between the original topology and the projected one.

The authors presented mathematical bases and a practical and computational vision of the process. UMAP is described as steps that construct and manipulate a weighted graph in the latter. The first step creates a graph, considering the weight function as

$$w(x_i, x_{ij}) = \exp\left(\frac{-\max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i} \right), \tag{32}$$

$\rho_i = \min\{d(x_i, x_{ij}) | 1 \leq j \leq k, d(x_i, x_{ij}) > 0\}$ and σ_i being calculated to satisfy:

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i} \right) = log_2(k). \tag{33}$$

These formulations consider a parameter k and some similarity function d. Thus, the not directed graph G used by UMAP has an adjacent matrix $B = A + A^T - A \circ A^T$, being A

a matrix generated with the previous weight function and the operator ∘ of the Hadamard product [79].

UMAP applies a Force Directed approach to place the graph nodes in the projected space. The algorithm uses attraction and repulsion functions, respectively, defined as

$$\frac{-2ab\|\mathrm{y_i}-\mathrm{y_j}\|_2^{2(b-1)}}{1+\|\mathrm{y_i}-\mathrm{y_j}\|_2^2}w(x_i,x_j)(\mathrm{y_i}-\mathrm{y_j}) \quad \text{and} \tag{34}$$

$$\frac{2b}{(\epsilon+\|\mathrm{y_i}-\mathrm{y_j}\|_2^2)(1+a\|\mathrm{y_i}-\mathrm{y_j}\|_2^{2b})}(1-w(x_i,x_j)(\mathrm{y_i}-\mathrm{y_j})), \tag{35}$$

where a and b are hyperparameters, with ϵ being a small number to avoid division by zero (the authors used $\epsilon = 0.001$). McInnes et al. [29] highlighted that the initial projection could be randomly generated; however, they used a spectral layout to initialize the projection. As a result, it provides faster convergence and superior algorithm stability.

In the last projection step, UMAP minimizes with gradient descent a smooth approximation of the strength between two elements (the original data instance and the projected one), which is defined as $\Phi(\mathrm{x},\mathrm{y}) = (1 + a(\|\mathrm{x}-\mathrm{y}\|_2^2)^b)^{-1}$. The parameters a and b are defined with a nonlinear least-square that fits one function dependent on the original instance and projected point position and a parameter that defines the desired separation of the points on transformed space.

As stated by its authors, UMAP provides highly similar visual results to t-SNE, and for this reason, it is often seen as a modern, faster, and more scalable alternative to t-SNE, providing visually similar outputs [56]. Figure 3 depicts a comparison between UMAP and t-SNE applied to well-known datasets, namely COIL20 [80], MNIST [81], Fashion MNIST [82], and Google News [83]. One can figure out that UMAP results are comparable with t-SNE ones, segregating and maintaining the same data structures.

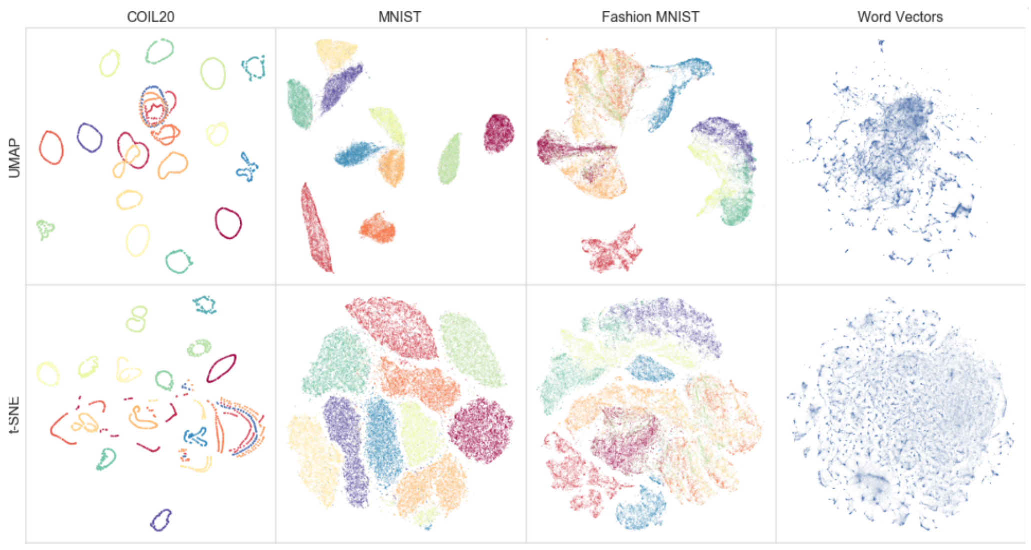

Figure 3. Comparing UMAP and t-SNE applied to some well-known datasets. Point colors represent data labels in each dataset. Source: McInnes et al. [29].

4.14. TopoMap

Following the topological approach, Doraiswamy et al. [56] developed TopoMap, a projection method devoted to mapping data from a high-dimensional space to a visual space preserving the global 0-dimensional topological persistence diagram defined by a Rips filtration over the m-dimensional data. Furthermore, TopoMap ensures that Rips filtrations generate the same connected components when applied to the original and the projected data by preserving the minimal spanning tree (MST) of the original data embedding in the projection [84].

Geometric relations such as distances or the proximity relationship between data instances are not the only attractive property to be preserved in a projection. According to TopoMap's authors, particular structures such as clusters and outliers could be more reliable and meaningful if the mapping reveals some topological invariants such as connected components and loops.

In contrast to dissimilarity-based projections, TopoMap is based on the non-differentiable 0-homology topological persistence evolution of cycles in the simplicial complexes resulting from a Euclidean distance-based Rips filtration [85]. Given a set of m-dimensional data instances P, TopoMap builds a complete weighted graph over P, weighting each edge (p_i, p_j) with the Euclidean distance between the corresponding endpoints, $d(p_i, p_j)$. The Rips filtration [86] grows a high-dimensional ball around the data instances, adding an edge (simplexes) to the filtration, one at a time, when two (or more) balls intersect each other. The addition of each new simplex can change the topology. The ordered set of changing edges is the MST of edges computed over the graph. TopoMap projects the data while preserving the evolution of cycles, such that the topological filtrations over the high-dimensional and projected data generate the same connected components (0-cycle) at the same instances of the respective filtrations.

TopoMap is a different approach from Isomap and UMAP. Isomap does not consider 0-homology groups, which are therefore not preserved. UMAP is based on category theory [29] and TopoMap focuses on persistent homology. Topological methods have been used to evaluate the projection process [87]. Since TopoMap bears theoretical guarantees, a side contribution is that it can be applied as an analytical tool helping to probe and illustrate how other projection methods behave, especially regarding distortions and cluster preservation. In this context, Doraiswamy et al. [56] demonstrated that clusters visualized in a t-SNE layout, in fact, tend to correspond only to pieces of clusters present in the m-dimensional data.

TopoMap can be used with an alternative distance metric than Euclidean distance. However, changing the metric requires computing the MST using this new metric, which means the running time for computing the MST will degenerate TopoMap complexity to $O(n^2)$ due to the computation of the distance matrix.

4.15. Graph Regularization Multidimensional Projection

The recently introduced Graph Regularization Multidimensional Projection (GRMP) [77] generates two-dimensional visualizations (2D) that reproduce the groups present in the high-dimensional space while preserving distance relationships between dataset instances. In addition, based on a similarity graph, GRMP uses the Graph Signal Processing theory (GSP) [88] and graph regularization as fundamental tools to incorporate these groups' information into the projected space.

A graph is a structure that represents data instances (vertices) and their relationships (edges). Signal theory adds extra information to the graph allowing the processing of a signal through the graph structure. In this context, graph regularization is based on signal smoothing by solving a minimization problem, which is equivalent to applying a low-pass spectral filter [77].

The GRMP method can be divided into three main steps. The initial step is related to the construction of the similarity graph. Similar to the previous topological-based methods, GRMP builds a similarity graph from the m-dimensional dataset matching data instances as the graph vertices set, establishing a one-to-one correspondence between instances and vertices. Every two vertices are connected according to the similarity relationship between their corresponding instances. In the second step, GRMP defines the coordinate graph signals in the 2D space through the spreading control given by a phyllotactic distribution. Note that the method moves away from all points using this distribution and only approximates the points that should be close in the final step, using regularization. In the third and last step, the coordinate graph is seen as signals to be processed and regularized using GSP tools.

GPS application is motivated because the neighbor points in the 2D space tend to have similar coordinates in the high-dimensional space. The smoothing effect of the graph regularization resets the coordinates of the points given by a phyllotactic distribution. The final GRMP projection is allocated in the visual space using control points selected from the medoids of each component. The control points are projected using MDS or other multidimensional projection techniques that preserve distances. Then, the smoothed components are translated and positioned, matching the control points' projection counterparts.

GRMP uses the k-NN graph [89] to build the similarity graph, which means GRMP inherits its main properties and difficulties in dealing with outliers. Parameterization is simplified in GRMP, comprising essentially only two parameters: the number of neighbors and the graph regularization parameter. However, the authors based their parameter choices on empirical tests referring to the need for fine-tuning to achieve better groupings.

4.16. SHAP Clustering

A fundamental challenge in the multidimensional projection domain is related to metric spaces. As we have seen through this section, most projection methods rely on the concept of distance to build similarities relationships between instances that will embed the high-dimensional space in the visual one. However, in projection and clustering tasks, we often need to handle data with features bearing very different types, i.e., features may be unitless scores, grams, meters, and others. Using features with different types as dimensions in a single multidimensional space forces any distance metric to compare the relative relationship in different units (grams vs. meters, for example) [90].

To tackle this issue, Lundberg et al. [90] applied SHAP values to convert the input features into importance values with the same units. SHAP (SHapley Additive exPlanations) is a computationally efficient way to verify the relationship between input features and the output of supervised machine learning methods. Specifically, SHAP is a feature importance model-agnostic method applied in the context of Explainable Artificial Intelligence (XAI) [91] to explain machine learning predictions through effect measurements reflecting the importance of input features. It is based on Shapley values [92] cost-sharing problem from classical cooperative game theory to share and allocate each feature a fair importance value for a particular prediction. The explicit definition is:

$$\phi_i = \sum_{S \subseteq m \setminus \{i\}} \frac{|S|!(m - |S| - 1)!}{m!} (f(S \cup \{i\}) - f(S)) \tag{36}$$

where $S \subseteq \{1, \ldots, m\}$ is a feature subset of all m features, $f(S \cup \{i\})$ and $f(S)$ represents a mapping function (a trained machine learning model, for example) with the feature i present and withheld, respectively. Thus, the Shapley value of a feature i can be understood as a weighted average of i's marginal contributions to every possible subset of features [93], which means Shapley values exact computation is an NP-hard problem unfeasible to be applied over high-dimensional data. Then, Lundberg and Lee [94] proposed SHAP as an approximation of classical Shapley values based on statistical samplings. SHAP is currently a state-of-the-art XAI method used to explain machine learning predictions in the more variate domains, ranging from aerospace to medicine and social sciences [95–97]. Lundberg et al. [90] extended SHAP to unsupervised learning context presenting SHAP Clustering.

SHAP Clustering converts all input features into SHAP values, unitless measures of feature importance. Moreover, even when all original inputs are in the same units, often some of them are more important than others. SHAP Clustering also introduces the concept of "supervised clustering" as a consequence of converting input features into SHAP values to obtain their effect information. In other words, SHAP values are log-odds attributing the impact of each feature to a machine learning model, then fluctuations in the feature values only affect the clustering if those fluctuations have significance to the outcome [90].

SHAP Clustering differs significantly from everything in previous multidimensional projection literature by applying concepts from game theory, positioning itself as an interesting and little-explored alternative. Moreover, SHAP Clustering handles one challenging issue in the multidimensional projection domain allowing direct comparisons between the relative importance of features bearing the more different units. Similarly, a recent approach used Shapley values as a tool to provide explanations regarding important features of the clustering process of multidimensional projection methods [98].

In the next section, we will discuss a comparative summary of the approaches reviewed in this section.

5. Discussions about the Multidimensional Projection Techniques

A challenging issue is a balance between projection quality and the required processing amount to generate good mappings. The high computational costs virtually block the application of some projection techniques in medium-sized datasets, even for the accurate techniques. As we saw above, an alternative strategy employed to mitigate the computational bound and enable the use of multidimensional projections, even in interactive and real-time applications, is the sampling of representative instances. From a subset of samples, it is possible to apply a precise (and more expensive) projection technique to generate the control points, which will guide the remaining mapping (now using a less expensive technique) in the visual space.

Although the original purpose of control points is to reduce computational time, control points can also be applied to incorporate some degree of user interactivity in projection processes [64]. However, there are still no comprehensive studies that quantify the impact introduced in manipulating control points by users, either in terms of quality or reasonability of generated maps [7]. Furthermore, the proper choice of representative samples is a central question in some circumstances. For example, authors such as Paulovich et al. [9], Paulovich and Minghim [10], Joia et al. [12], and Fadel et al. [6] suggested a random selection of large quantities of instances due to the difficulty in determining an ideal amount that effectively represents all the classes contained in the dataset, but exactly how large this must be this quantity is an open issue.

The larger the representative subset, the greater the computational effort required in the initial projection stage. Although it is statistically possible to infer that a random sample contains significant information from the dataset, there is no guarantee that it will always be true. Furthermore, the random selection introduces an undesirable effect: the same dataset is mapped into different layouts, something that can generate some confusion for the users. Thus, there is a need for an alternative to reducing the complexity of multidimensional projection techniques while producing effective and deterministic layouts in a reduced time.

Figure 4 presents a performance comparison between some of the state-of-the-art multidimensional projection techniques. According to Fadel et al. [6], global techniques usually yield better dissimilarity preservation averages (stress) and computational performance. Note that LAMP, which is essentially a local method, is highlighted as one of the best balances between stress (see Figure 4 left-hand) and computational performance (see Figure 4 right-hand).

Table 1 summarizes the main approaches reviewed throughout the text. In addition to these techniques, there are other derivations, modifications, and improvements. Maaten et al. [16] also carried out a valuable review of the subject. However, the interest in developing and improving multidimensional projection strategies has been intense over the past ten years. In this scenario, Nonato and Aupetit [7] carried out comprehensive and updated research about multidimensional projection methods, clarifying limitations and strengths from the perspective of visual analytics.

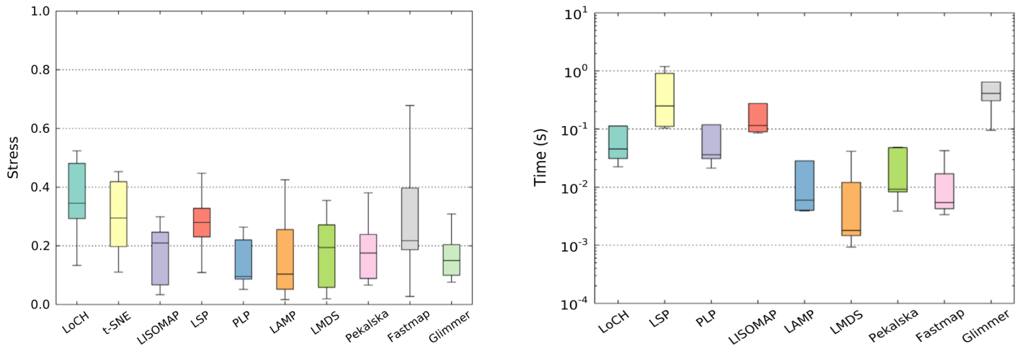

Figure 4. Comparative charts presenting stress and running times between some multidimensional projection techniques. Source: Fadel et al. [6].

As we are demonstrating here, the multidimensional projection has a large field of application possibilities. To finish this Discussion section, we highlight one emerging application in a new effervescent research domain: Explainable Artificial Intelligence (XAI). Artificial Intelligence has a significant growing development interest in the last decade, especially Machine Learning algorithms. On the one hand, Machine Learning models such as those based on Artificial Neural Networks and Gradient Boosting can outperform human capabilities in many tasks and, for this reason, have been often applied in decision-making processes [99,100]. On the other hand, these kinds of Machine Learning models are based on high-complex structures, which makes them "black boxes" virtually impossible to be interpreted, which raises several issues regarding trust in their predictions when applied to sensitive contexts, e.g., medicine, finances, autonomous cars, and so on [91].

XAI methods are designed to "open" the Machine Learning black boxes providing information to justify and understand the work logic of these algorithms. An interesting and few explored use of multidimensional projections is in the XAI context. We can cite applications of t-SNE as part of XAI proposes aiming to cluster images according to neuron activation in networks [101], to visualize latent variables [102], clustering genetic variables [103], and the above-mentioned SHAP Clustering. XAI is a challenging subject, and we argue that visualization approaches such as multidimensional projections can improve explainability systems because visualization can produce high-informative and human-centered explanations [104]. The interested reader can consult Arrieta et al. [91] and Adadi and Berrada [104] surveys for a detailed view of XAI theory.

5.1. Real Data Applications

In order to illustrate the performance of some of the exposed multidimensional projection algorithms, two different applications were chosen: The clothing E-Commerce text mining task and the DNA microarray Cancer Classification task. It is essential to mention that some embedding techniques are based on distance metrics, such as Euclidean, Manhattan, Cosine, Minkowski, Jaccard, and Chebyshev. Nonetheless, for illustration, we adopted only the Euclidean distance based on the Classical MDS, Force Scheme, LAMP, t-SNE, and UMAP; in addition, compared to the PCA, which used the observation value itself. The first example was solved using R software [105] and the second through Python [106].

5.1.1. Text Mining Data

The first motivation was adopted from the KAGGLE website, which is related to the Women's Clothing E-Commerce database. These data were anonymous recorded reviews derived from customers, which its dataset shows ten features (Clothing ID, reviewers Age, Title, Review Text, Rating, Recommended IND, Positive Feedback Count, Division Name, Department Name, and Class Name). Further details can be found at [107].

After pre-processing steps such as removing numbers, punctuation, and stopwords and transforming them all into lowercase, the frequency of the terms was calculated.

Figure 5 shows the description of the Review Text feature, a suitable analysis that helps verify words' importance and their relation with the recommendations.

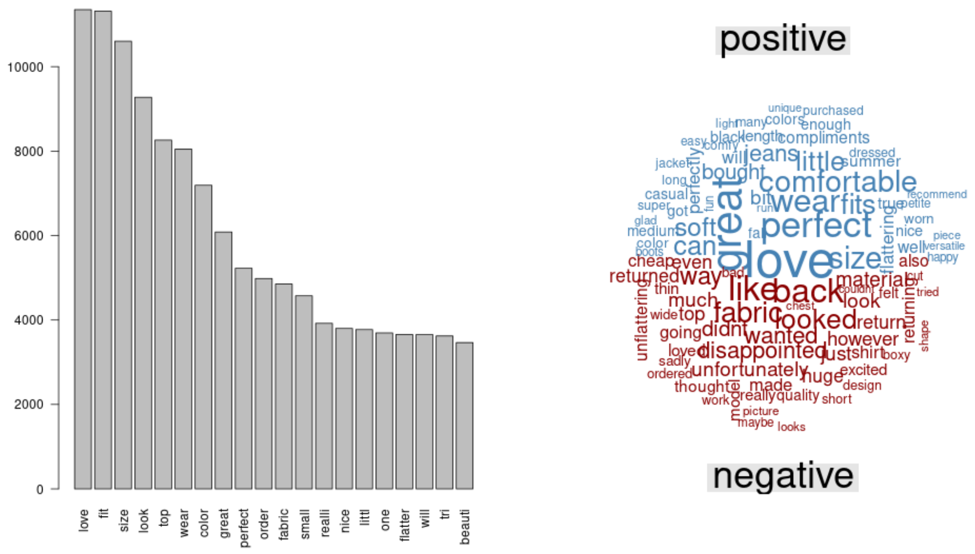

Figure 5. Text summary: the 20 words more frequent (left), and a word cloud of 100 terms colored by the recommended (positive or negative).

In addition to these analyses, the multidimensional projection algorithms can cluster observations alike embedded in 2D. Figure 6 shows the MP exemplification considering the review from the item Jacket only, adopting the algorithms PCA, MDS, Force Scheme, LAMP, t-SNE, and UMAP, whereas the blue words are used in positive comments, and the red used in negative comments. Whereas these engines are helpful in monitoring and tracing some characteristics that can maximize the rating from the clothing store. In this case, it is challenging to identify clusters between positive and negative comments. Thus, other types of features could be considered to represent that segregation. On the other hand, one can analyze isolated clothing classes to determine the segregation between positive and negative comments.

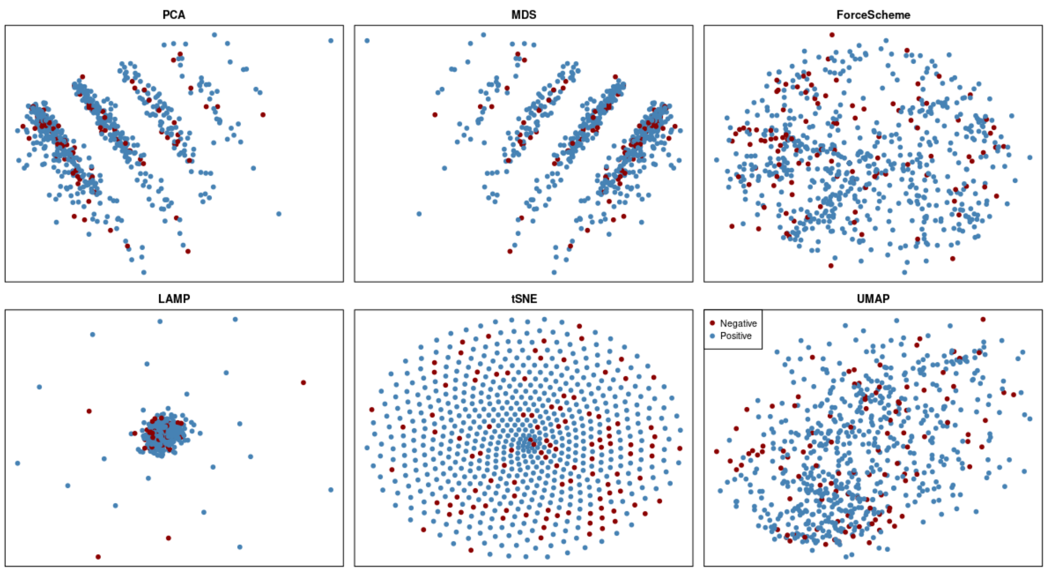

Figure 6. Projections of the user reviews, colored by the recommended labels (negative or positive). Source: elaborated by the authors.

For instance, by selecting PCA visualization and adopting the package *factoextra*, the text reviews can be seen through their similarity (in Figure 7). Selecting the extreme values on the top right (ID #8551) versus the bottom right (ID #12374), it is easy to detect we have a positive review versus a negative one. Moreover, those obtained patterns can unravel other characteristics across the formed clusters.

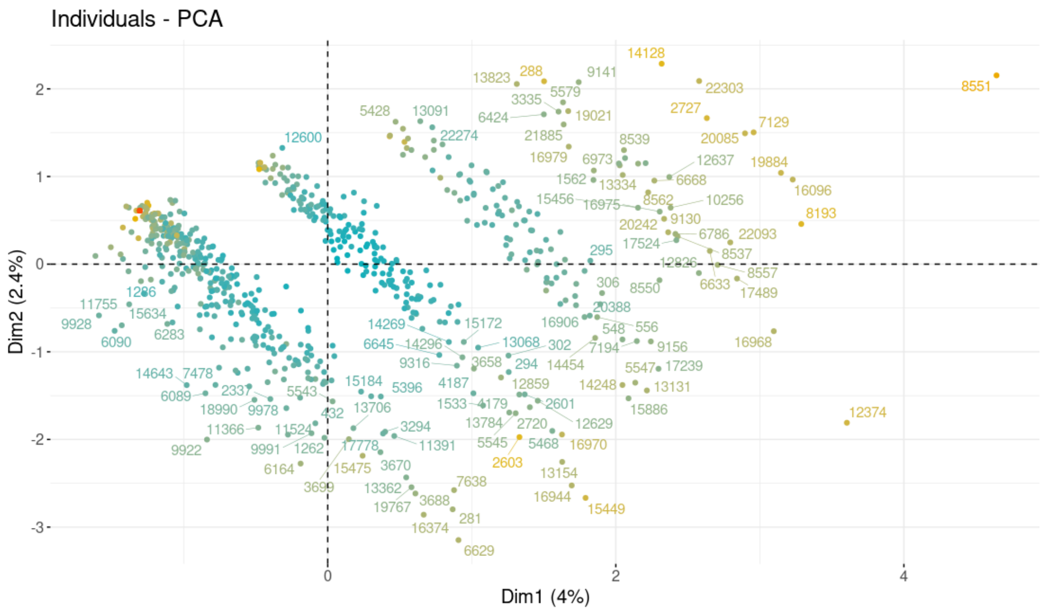

Figure 7. PCA of the reviews related to Jacket item. Numbers around points represent review IDs. The individual's color (point) was randomly allocated according to their qualities of representation. Source: elaborated by the authors.

Review ID #8551—"This brand makes the best jackets. I got a vest from them last year and an olive moto jacket. Both fit amazing. This one is no different. It fits great! I love that it is a great casual jacket with structure and drape. I cannot say enough good things about this Jacket. Every time I wear it, I feel so pulled together. Fun light gray color that you do not see too often in jackets. This is a jacket I will see in my wardrobe for years to come. must-have".

Review ID #12374—"This is a very loose-fitting style jacket. If you like your jackets oversized, get your regular size; if you prefer the slightly loose (but not baggy) fit like the product photo, size down. As noted by another reviewer, the sleeves are incredibly long. I would say if you are a typical size 0 or 2 (unless you like the oversized fit), you are likely sized out of this Jacket. It is a shame because the quality is great, and the windowpane print is beautiful. I am sadly sized out".

5.1.2. Bio-Informatics (Cancer Classification Data)

Golub et al. [108] discussed gene expression monitoring (via DNA microarray) related to patients with acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). This dataset targets identifying new cancer classes and assigning tumors to known classes, though it deals with high-dimensional data. Dataset and further details can be found at [109].

The adopted dataset presents 7129 observations, showing 78 gene expressions (features). Figure 8 shows the correlation matrix across the 78 gene expressions.

Moreover, Figure 9 presents six multidimensional projection techniques (PCA, MDS, Force Scheme, LAMP, t-SNE, UMAP), implemented in Python, to embed characteristics in two dimensions. The specialist can combine those finds derived and combined from all of them. One can say those projections can be complementary, and by using visual data mining, hidden patterns may be found.

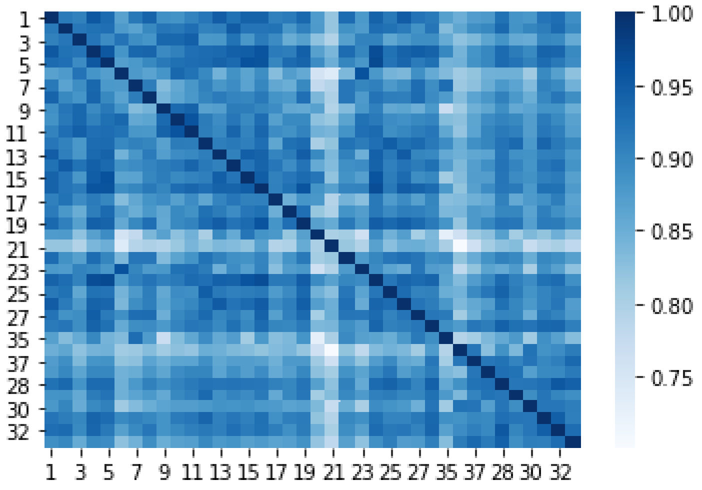

Figure 8. Correlation matrix across the 78 gene expressions (DNA microarray). Source: elaborated by the authors.

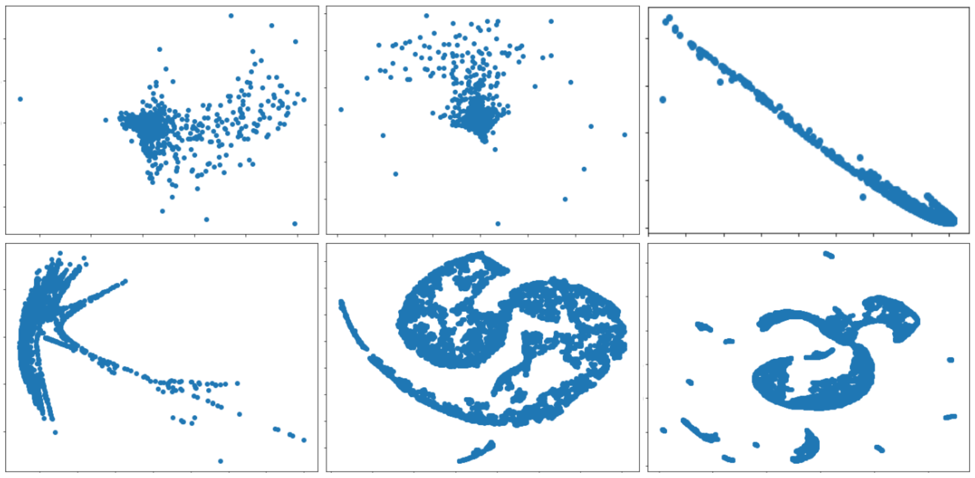

Figure 9. Projections of the DNA micro-array, represented by 78 gene expressions characteristics. Source: elaborated by the authors.

6. Challenges, Open Questions, and Future Research Directions

Given visual analytics' relevance as a significant part of mining tools in data science applications, multidimensional projection techniques are a powerful resource due to their ability to generate visual representations of complex data instances groupings belonging to high-dimensional spaces. However, significant gaps still need to be addressed in the multidimensional projection context.

The trade-off between computational costs and precision is one of the significant challenges regarding the development and application of multidimensional projections. Because of the increasingly large datasets we are facing nowadays, both in terms of the number of instances and the number of attributes per instance (dimensionality), the development of computationally efficient and accurate techniques is a challenging issue that involves research in sophisticated mathematical solutions. This challenge is intensified when applications need to handle real-time or growing datasets, such as time series and streaming data.

An open opportunity in the projection domain is to handle time-varying data, i.e., how to project multidimensional time series. Research in multidimensional time series has a long and rich history [110], but properly exploring the time component imposes a major challenge in the projection domain. Though it is possible to project time slices of the data and analyze the evolution of these time slices, it can easily become an overwhelming task when a long time series is considered. The literature is not so rich regarding multidimensional projection methods able to capture dynamic data evolving [111]. We can cite some extensions of well-known methods such as t-SNE [112–114] and even PCA [115,116].

Real-time data also highlights an issue already identified by the research community in multidimensional projections: stability. A stable method must be able to incorporate new data in the visual space without significant modifications in the position of points already projected. In addition, in a scenario with continuously generated data, it might be too expensive to recalculate the entire projection so that the number of instances is increased/decreased. Methods that rely on matrix decomposition are not stable since the addition/removal of a single instance demands the recalculation of all matrix operations, which can significantly change the mapping [7]. Optimization-based techniques depend on initial conditions, so there is no way to guarantee their stability.

Control points-based strategies are more resilient alternatives to perturbations, as long as the control points are kept fixed. However, these kinds of approaches open other questions. As mentioned before, the random sampling of control points has been a commonly applied solution. However, there is no guarantee that a random sample encompasses all the nuances of a dataset (besides may render some confusion when generating different layouts). Another issue we raised in this context is related to the sampling size. How many control points are needed to compose a good sample? Thus, these open questions are waiting for future research that establishes metrics and parameters for the generation of the initial sample subsets.

Most of the multidimensional projection techniques were tailored to operate on numerical data. Thus, we identified a gap between projection tools in their ability to handle non-numerical values, such as ordinal, textual, categorical data [117], and so on. For example, one-hot-encoding is a well-known method applied for a long time in digital circuits and machine learning environments that can be used in our context to transform categorical variables into numerical combinations of binary nature. However, there is still work to properly deal with non essentially numerical variables, although some classic strategies can be applied to alleviate this deficiency, such as one-hot-encoding.

One major challenge of multidimensional projection techniques, in our opinion, is interpretability. In practical terms, there is a lack of information transmission when a user with little or no additional training analyzes the result of a multidimensional projection mapping. Although some studies have already worked on improving scatter plots' perceptual capabilities [118–121], only the presentation of points or clouds of points into scatter plots does not effectively inform the user about the complex relationships between data instances. What features on the m-dimensional space lead to the produced mapping on the p-dimensional space? As we can see, there is a demand for layouts with designs that properly explore the users' graphic perception abilities, enriched with statistical information, and interactive tools that enable the user to conduct an exploratory visual analysis. There is a clear need for enrichment of multidimensional projection mappings, which could lead users to generate insights over the available information, transforming the perception of point distances on the visual space into the decoding of the original (dis)similarities between data instances.

7. Conclusions

In this study, we discussed the main multidimensional projection techniques and, whenever possible, their more applied derivations. Our taxonomy and formalization include definitions not only of the covered techniques of this research but also seminal concepts about application domains, the pre-processing data tasks, and the limitations that multidimensional projection techniques addresses. Furthermore, the ability to preserve data similarity while generating visual representations renders multidimensional projection techniques highly informative as a visualization mechanism that can enrich many problem-solving and decision-making applications [122,123]. Thus, we believe that multidimensional projection is an effervescing research area with a considerable applicability potential that comes to contribute to the existing exploratory tools in a wide range of domains within the data sciences. Nevertheless, many important issues are still open to be explored, and we pointed out several scenarios and directions where multidimensional techniques can evolve in future works.

Author Contributions: Conceptualization, methodology, software, validation, data curation, writing–original draft preparation, visualization, E.S.O., F.F.D., and D.C.d.N.; funding acquisition, D.C.d.N. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. The views expressed are those of the authors and do not reflect the official policy or position of the sponsor.

Informed Consent Statement: Not applicable.

Data Availability Statement: Further details regarding data support and scripts in R and Python programming languages can be found at https://github.com/ProfNascimento/MP (accessed on 20 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Javed, W.; McDonnel, B.; Elmqvist, N. Graphical Perception of Multiple Time Series. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 927–934. [CrossRef] [PubMed]
2. Shneiderman, B. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proceedings of the 1996 IEEE Symposium on Visual Languages, VL'96, Boulder, CO, USA, 3–6 September 1996; IEEE Computer Society: Washington, DC, USA, 1996; p. 336.
3. Heer, J.; Kong, N.; Agrawala, M. Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI'09, Boston, MA, USA, 4–9 April 2009; ACM: New York, NY, USA, 2009; pp. 1303–1312.
4. Chi, E.H. *A Framework for Visualization Information*; Springer: Berlin/Heidelberg, Germany, 2002.
5. Telea, A.C. *Data Visualization: Principles and Practice*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2014.
6. Fadel, S.G.; Fatore, F.M.; Duarte, F.S.L.G.; Paulovich, F.V. LoCH: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing* **2015**, *150*, 546–556. [CrossRef]
7. Nonato, L.G.; Aupetit, M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 2650–2673. [CrossRef] [PubMed]
8. Cantareira, G.D.; Etemad, E.; Paulovich, F.V. Exploring Neural Network Hidden Layer Activity Using Vector Fields. *Information* **2020**, *11*, 426. [CrossRef]
9. Paulovich, F.V.; Nonato, L.G.; Minghim, R.; Levkowitz, H. Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE Trans. Vis. Comput. Graph.* **2008**, *14*, 564–575. [CrossRef] [PubMed]
10. Paulovich, F.V.; Minghim, R. HiPP: A Novel Hierarchical Point Placement Strategy and Its Application to the Exploration of Document Collections. *IEEE Trans. Vis. Comput. Graph.* **2008**, *14*, 1229–1236. [CrossRef]
11. Paulovich, F.V.; Moraes, M.L.; Maki, R.M.; Ferreira, M.; Oliveira, O.N.; de Oliveira, M.C.F. Information visualization techniques for sensing and biosensing. *Anal. R. Soc. Chem.* **2011**, *136*, 1344–1350. [CrossRef]
12. Joia, P.; Coimbra, D.; Cuminato, J.A.; Paulovich, F.V.; Nonato, L.G. Local Affine Multidimensional Projection. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 2563–2571. [CrossRef]
13. Berkhin, P. *A Survey of Clustering Data Mining Techniques*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71.
14. Lee, J.; Verleysen, M. *Nonlinear Dimensionality Reduction*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
15. Buja, A.; Swayne, D.F.; Littman, M.L.; Dean, N.; Hofmann, H.; Chen, L. Data Visualization With Multidimensional Scaling. *J. Comput. Graph. Stat.* **2008**, *17*, 444–472. [CrossRef]
16. Maaten, L.V.D.; Postma, E.; Herik, J.V.D. Dimensionality Reduction: A Comparative Review. *J. Mach. Learn Res.* **2009**, *10*, 13.
17. Osipyan, H.; Kruliš, M.; Marchand-Maillet, S. A Survey of CUDA-based Multidimensional Scaling on GPU Architecture. In Proceedings of the 2015 Imperial College Computing Student Workshop (ICCSW 2015), London, UK, 24–25 September 2015; OpenAccess Series in Informatics (OASIcs); Schulz, C., Liew, D., Eds.; Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2015; Volume 49, pp. 37–45.
18. 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 Trans. Vis. Comput. Graph.* **2017**, *23*, 241–250. [CrossRef]

19. Konyha, Z.; Lez, A.; Matković, K.; Jelović, M.; Hauser, H. Interactive Visual Analysis of Families of Curves Using Data Aggregation and Derivation. In Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, i-KNOW'12, Graz, Austria, 5–7 September 2012; ACM: New York, NY, USA, 2012; pp. 24:1–24:8.
20. Fua, Y.H.; Ward, M.O.; Rundensteiner, E.A. Hierarchical Parallel Coordinates for Exploration of Large Datasets. In Proceedings of the Conference on Visualization'99: Celebrating Ten Years, VIS'99, Los Alamitos, CA, USA, October 1999; IEEE Computer Society Press: Los Alamitos, CA, USA, 1999; pp. 43–50.
21. Ware, C. *Information Visualization: Perception for Design*, 2nd ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2004.
22. Pudil, P.; Novovicova, J. Novel Methods for Subset Selection with Respect to Problem Knowledge. *IEEE Intell. Syst.* **1998**, *13*, 66–74. [CrossRef]
23. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
24. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
25. Kirby, M. *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*, 1st ed.; John Wiley & Sons, Inc.: New York, NY, USA, 2000.
26. Tejada, E.; Minghim, R.; Nonato, L.G. On Improved Projection Techniques to Support Visual Exploration of Multidimensional Data Sets. *Inf. Vis.* **2003**, *2*, 218–231. [CrossRef]
27. Cox, T.F.; Cox, M.A.A. *Multidimensional Scaling*, 2nd ed.; Chapman and Hall–CRC: New York, NY, USA, 2000.
28. Maaten, L.V.D.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
29. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
30. Zezula, P.; Amato, G.; Dohnal, V.; Batko, M. *Similarity Search: The Metric Space Approach*, 1st ed.; Advances in Database Systems, Book 32; Springer Publishing Company, Inc.: New York, NY, USA, 2006.
31. Paulovich, F.V. Mapeamento de Dados Multi-Dimensionais-Integrando Mineração e Visualização. Ph.D. Thesis, Universidade de São Paulo, São Paulo, Brazil, 2008.
32. Ward, M.O.; Grinstein, G.; Keim, D. *Interactive Data Visualizaton: Foundations, Techniques, and Applications*, 2nd ed.; CRC Press: New York, NY, USA, 2015; p. 578.
33. Keogh, E.; Kasetty, S. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Data Min. Knowl. Discov.* **2003**, *7*, 349–371. [CrossRef]
34. Law, M.H.C.; Jain, A.K. Incremental Nonlinear Dimensionality Reduction by Manifold Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 377–391. [CrossRef]
35. Kruskal, J.B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **1964**, *29*, 1–27. [CrossRef]
36. Heulot, N.; Fekete, J.D.; Aupetit, M. Visualizing dimensionality reduction artifacts: An evaluation. *arXiv* **2017**, arXiv:1705.05283.
37. Rousseeuw, P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
38. Coimbra, D.B. Multidimensional Projections for the Visual Exploration of Multimedia Data. Ph.D. Thesis, Universidade de São Paulo, São Paulo, Brazil, 2016.
39. Marcilio, W.E.; Eler, D.M.; Garcia, R.E. An approach to perform local analysis on multidimensional projection. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 17–18 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 351–358.
40. Bertini, E.; Tatu, A.; Keim, D. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 2203–2212. [CrossRef] [PubMed]
41. Cleveland, W.S.; McGill, R. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J. Am. Stat. Assoc.* **1984**, *79*, 531–554. [CrossRef]
42. Wagemans, J.; Elder, J.H.; Kubovy, M.; Palmer, S.E.; Peterson, M.A.; Singh, M.; Heydt, R.V.D. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychol. Bull.* **2012**, *138*, 1172–1217. [CrossRef]
43. Becker, R.A.; Cleveland, W.S. Brushing Scatterplots. *Technometrics* **1987**, *29*, 127–142. [CrossRef]
44. Alexandrina, E.C.; Ortigossa, E.S.; Lui, E.S.; Gonçalves, J.A.S.; Corrêa, N.A.; Nonato, L.G.; Aguiar, M.L. Analysis and visualization of multidimensional time series: Particulate matter (PM_{10}) from São Carlos-SP (Brazil). *Atmos. Pollut. Res.* **2019**, *10*, 1299–1311. [CrossRef]
45. McLachlan, P.; Munzner, T.; Koutsofios, E.; North, S. LiveRAC: Interactive Visual Exploration of System Management Time-series Data. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'08, Florence, Italy, 5–10 April 2008; ACM: New York, NY, USA, 2008; pp. 1483–1492.
46. Jugel, U.; Jerzak, Z.; Hackenbroich, G.; Markl, V. M4: A Visualization-oriented Time Series Data Aggregation. *Proc. Very Large Database Endow.* **2014**, *7*, 797–808. [CrossRef]
47. Jolliffe, I. *Principal Component Analysis*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2002.
48. Li, L.; Su, X.; Zhang, Y.; Lin, Y.; Li, Z. Trend Modeling for Traffic Time Series Analysis: An Integrated Study. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3430–3439. [CrossRef]
49. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2022.

50. Torgerson, W.S. Multidimensional scaling: I. Theory and method. *Psychometrika* **1952**, *17*, 401–419. [CrossRef]
51. Silva, V.D.; Tenenbaum, J.B. *Sparse Multidimensional Scaling Using Landmark Points*; Technical Report; Stanford University: Stanford, CA, USA, 2004.
52. Brandes, U.; Pich, C. Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. In Proceedings of the 14th International Conference on Graph Drawing, GD'06, Tübingen, Germany, 18–20 September 2007; Springer: Berlin, Heidelberg, 2007; pp. 42–53.
53. Sammon, J.W. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.* **1969**, *18*, 401–409. [CrossRef]
54. Pekalska, E.; de Ridder, D.; Duin, R.P.; Kraaijveld, M.A. A new method of generalizing Sammon mapping with application to algorithm speed-up. In Proceedings of the 5th Annual Conference of the Advanced School for Computing and Imaging, ASCI'99, Heijen, The Netherlands, 15–17 June 1999; Delft: Heijen, The Netherlands, 1999; pp. 221–228.
55. Tenenbaum, J.B.; Silva, V.d.; Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **2000**, *290*, 2319–2323. [CrossRef] [PubMed]
56. Doraiswamy, H.; Tierny, J.; Silva, P.J.; Nonato, L.G.; Silva, C. TopoMap: A 0-dimensional homology preserving projection of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 561–571. [CrossRef] [PubMed]
57. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
58. Balasubramanian, M.; Schwartz, E.L. The Isomap Algorithm and Topological Stability. *Science* **2002**, *295*, 7a. [CrossRef] [PubMed]
59. Silva, V.D.; Tenenbaum, J.B. Global Versus Local Methods in Nonlinear Dimensionality Reduction. In Proceedings of the 15th International Conference on Neural Information Processing Systems, NIPS'02, Vancouver, BC, Canada, 9–14 December 2002; MIT Press: Cambridge, MA, USA, 2002; pp. 721–728.
60. Faloutsos, C.; Lin, K.I. FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets. *ACM SIGMOD Rec.* **1995**, *24*, 163–174. [CrossRef]
61. Chalmers, M. A Linear Iteration Time Layout Algorithm for Visualising High-dimensional Data. In Proceedings of the 7th Conference on Visualization'96, VIS '96, San Francisco, CA, USA, 27 October–1 November 1996; IEEE Computer Society Press: Los Alamitos, CA, USA, 1996; p. 127.
62. Morrison, A.; Ross, G.; Chalmers, M. A Hybrid Layout Algorithm for Sub-Quadratic Multidimensional Scaling. In Proceedings of the IEEE Symposium on Information Visualization, INFOVIS'02, Boston, MA, USA, 27 October–1 November 2002; IEEE Computer Society: Washington, DC, USA, 2002; p. 152.
63. Brodbeck, D.; Girardin, L. Combining topological clustering and multidimensional scaling for visualising large data sets. In Proceedings of the IEEE Information Visualization 1998, Research Triangle Park, NC, USA, 19–20 October 1998; pp. 1–4.
64. Ingram, S.; Munzner, T.; Olano, M. Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Vis. Comput. Graph.* **2009**, *15*, 249–261. [CrossRef]
65. Hinton, G.; Roweis, S. Stochastic Neighbor Embedding. In Proceedings of the 15th International Conference on Neural Information Processing Systems, NIPS'02, Vancouver, BC, Canada, 9–14 December 2002; MIT Press: Cambridge, MA, USA, 2002; pp. 857–864.
66. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
67. Maaten, L.V.D. Accelerating t-SNE Using Tree-based Algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
68. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
69. Lee, J.A.; Verleysen, M. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Comput. Sci.* **2011**, *4*, 538–547. [CrossRef]
70. Wattenberg, M.; Viégas, F.; Johnson, I. How to use t-SNE effectively. *Distill* **2016**, *1*, e2. [CrossRef]
71. Paulovich, F.V.; Eler, D.M.; Poco, J.; Botha, C.P.; Minghim, R.; Nonato, L.G. Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization. In Proceedings of the 13th Eurographics/IEEE—VGTC Conference on Visualization, EuroVis'11, Bergen, Norway, 1–3 June 2011; The Eurographs Association and John Wiley & Sons, Ltd.: Chichester, UK, 2011; pp. 1091–1100.
72. Paulovich, F.V.; Silva, C.T.; Nonato, L.G. Two-Phase Mapping for Projecting Massive Data Sets. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 1281–1290. [CrossRef] [PubMed]
73. Gower, J.C.; Dijksterhuis, G.B. *Procrustes Problems;* Oxford Statistical Science Series; Oxford University Press: Oxford, UK, 2004; Volume 30.
74. Steinbach, M.; Karypis, G.; Kumar, V. A comparison of document clustering techniques. In Proceedings of the KDD Workshop on Text Mining, Boston, MA, USA, 20 August 2000; pp. 525–526.
75. Dias, F.; Minghim, R. xHiPP: eXtended Hierarchical Point Placement Strategy. In Proceedings of the 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Foz do Iguaçu, Brazil, 29 October–1 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 361–368.
76. Sturges, H.A. The choice of a class interval. *J. Am. Stat. Assoc.* **1926**, *21*, 65–66. [CrossRef]
77. Dal Col, A.; Petronetto, F. Graph regularization multidimensional projection. *Pattern Recognit.* **2022**, *129*, 108690. [CrossRef]
78. Goerss, P.G.; Jardine, J.F. *Simplicial Homotopy Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
79. Davis, C. The norm of the Schur product operation. *Numer. Math.* **1962**, *4*, 343–344. [CrossRef]
80. Nene, S.A.; Nayar, S.K.; Murase, H. *Columbia Object Image Library (COIL-20)*; Columbia University: New York, NY, USA, 1996.

81. LeCun, Y.; Cortes, C.; Burges, C.J. The MNIST Database of Handwritten Digits. 1998, Volume 10, p. 34. Available online: http://yann.lecun.com/exdb/mnist (accessed on 10 April 2022).
82. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
83. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
84. Wagner, A.; Solomon, E.; Bendich, P. Improving Metric Dimensionality Reduction with Distributed Topology. *arXiv* **2021**, arXiv:2106.07613.
85. Nelson, B.J.; Luo, Y. Topology-Preserving Dimensionality Reduction via Interleaving Optimization. *arXiv* **2022**, arXiv:2201.13012.
86. Bauer, U. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *J. Appl. Comput. Topol.* **2021**, *5*, 391–423. [CrossRef]
87. Sohns, J.T.; Schmitt, M.; Jirasek, F.; Hasse, H.; Leitte, H. Attribute-based Explanation of Non-Linear Embeddings of High-Dimensional Data. *IEEE Trans. Vis. Comput. Graph.* **2021**, *28*, 540–550. [CrossRef]
88. Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J.M.F.; Vandergheynst, P. Graph Signal Processing: Overview, Challenges, and Applications. *Proc. IEEE* **2018**, *106*, 808–828. [CrossRef]
89. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
90. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent individualized feature attribution for tree ensembles. *arXiv* **2018**, arXiv:1802.03888.
91. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
92. Shapley, L.S. A value for n-person games *Contributions to the Theory of Games (AM-28)*; Princeton University Press: Princeton, NJ, USA, 1953; Volume 2, pp. 307–318.
93. Molnar, C. *Interpretable Machine Learning*; Lulu Press, Inc.: Durham, NC, USA, 2019.
94. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 4768–4777.
95. Hong, C.W.; Lee, C.; Lee, K.; Ko, M.S.; Hur, K. Explainable Artificial Intelligence for the Remaining Useful Life Prognosis of the Turbofan Engines. In Proceedings of the 2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII), Taiwan, China, 21–23 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 144–147.
96. Lundberg, S.M.; Nair, B.; Vavilala, M.S.; Horibe, M.; Eisses, M.J.; Adams, T.; Liston, D.E.; Low, D.K.W.; Newman, S.F.; Kim, J.; et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nat. Biomed. Eng.* **2018**, *2*, 749–760. [CrossRef]
97. Vilarino, R.; Vicente, R. An Experiment on Leveraging SHAP Values to Investigate Racial Bias. *arXiv* **2020**, arXiv:2011.09865.
98. Marcílio, W.E., Jr.; Eler, D.M. Explaining dimensionality reduction results using Shapley values. *Expert Syst. Appl.* **2021**, *178*, 115020.
99. Chakraborty, S.; Tomsett, R.; Raghavendra, R.; Harborne, D.; Alzantot, M.; Cerutti, F.; Srivastava, M.; Preece, A.; Julier, S.; Rao, R.M.; et al. Interpretability of deep learning models: A survey of results. In Proceedings of the 2017 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
100. Zhang, Q.; Zhu, S.C. Visual interpretability for deep learning: a survey. *arXiv* **2018**, arXiv:1802.00614.
101. Nguyen, A.; Yosinski, J.; Clune, J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv* **2016**, arXiv:1602.03616.
102. Ma, W.; Cheng, F.; Xu, Y.; Wen, Q.; Liu, Y. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Adv. Mater.* **2019**, *31*, 1901111. [CrossRef]
103. Xu, W.; Jiang, X.; Hu, X.; Li, G. Visualization of genetic disease-phenotype similarities by multiple maps t-SNE with Laplacian regularization. *BMC Med Genom.* **2014**, *7*, S1. [CrossRef] [PubMed]
104. Adadi, A.; Berrada, M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
105. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2022.
106. Van Rossum, G.; Drake, F.L., Jr. *Python Tutorial*; Centrum Voor Wiskunde en Informatica: Amsterdam, The Netherlands, 1995.
107. Kaggle. Women's E-Commerce Clothing Reviews. 2018. Available online: https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews (accessed on 10 April 2022).
108. Golub, T.R.; Slonim, D.K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.P.; Coller, H.; Loh, M.L.; Downing, J.R.; Caligiuri, M.A.; et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **1999**, *286*, 531–537. [CrossRef] [PubMed]
109. Kaggle. *Gene Expression Classification*; Kaggle: San Francisco, CA, USA, 2019.

110. Shumway, R.H.; Stoffer, D.S. *Time Series Analysis and Its Applications: With R Examples*, 3rd ed.; Springer Publishing Company, Incorporated: New York, NY, USA, 2011.
111. Vernier, E.F.; Garcia, R.; Silva, I.d.; Comba, J.L.D.; Telea, A.C. Quantitative Evaluation of Time-Dependent Multidimensional Projection Techniques. *Comput. Graph. Forum* **2020**, *39*, 241–252. [CrossRef]
112. Rauber, P.E.; Falcão, A.X.; Telea, A.C. Visualizing Time-Dependent Data Using Dynamic t-SNE. In Proceedings of the Eurographics/IEEE VGTC Conference on Visualization, Groningen, The Netherlands, 6–10 June 2016.
113. Rauber, P.E.; Fadel, S.G.; Falcao, A.X.; Telea, A.C. Visualizing the hidden activity of artificial neural networks. *IEEE Trans. Vis. Comput. Graph.* **2016**, *23*, 101–110. [CrossRef] [PubMed]
114. Nguyen, M.; Purushotham, S.; To, H.; Shahabi, C. m-tsne: A framework for visualizing high-dimensional multivariate time series. *arXiv* **2017**, arXiv:1708.07942.
115. Mao, Y.; Dillon, J.; Lebanon, G. Sequential document visualization. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1208–1215. [CrossRef]
116. Fujiwara, T.; Chou, J.K.; Shilpika, S.; Xu, P.; Ren, L.; Ma, K.L. An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 418–428. [CrossRef]
117. Pereira, M.M.; Paulovich, F.V. RankViz: A visualization framework to assist interpretation of Learning to Rank algorithms. *Comput. Graph.* **2020**, *93*, 25–38. [CrossRef]
118. Gleicher, M.; Correll, M.; Nothelfer, C.; Franconeri, S. Perception of average value in multiclass scatterplots. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2316–2325. [CrossRef]
119. Rensink, R.A.; Baldridge, G. The perception of correlation in scatterplots. *Comput. Graph. Forum* **2010**, *29*, 1203–1210. [CrossRef]
120. Sedlmair, M.; Tatu, A.; Munzner, T.; Tory, M. A taxonomy of visual cluster separation factors. *Comput. Graph. Forum* **2012**, *31*, 1335–1344. [CrossRef]
121. Pandey, A.V.; Krause, J.; Felix, C.; Boy, J.; Bertini, E. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 3659–3669.
122. Chao, G.; Luo, Y.; Ding, W. Recent advances in supervised dimension reduction: A survey. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 341–358. [CrossRef]
123. Espadoto, M.; Martins, R.M.; Kerren, A.; Hirata, N.S.T.; Telea, A.C. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2153–2173. [CrossRef] [PubMed]