Spanning Trees as Approximation of Data Structures

Daniel Alcaide and Jan Aerts

Abstract—The connections in a graph generate a structure that is independent of a coordinate system. This visual metaphor allows creating a more flexible representation of data than a two-dimensional scatterplot. In this article, we present STAD (Simplified Topological Abstraction of Data), a parameter-free dimensionality reduction method that projects high-dimensional data into a graph. STAD generates an abstract representation of high-dimensional data by giving each data point a location in a graph which preserves the approximate distances in the original high-dimensional space. The STAD graph is built upon the Minimum Spanning Tree (MST) to which new edges are added until the correlation between the distances from the graph and the original dataset is maximized. Additionally, STAD supports the inclusion of additional functions to focus the exploration and allow the analysis of data from new perspectives, emphasizing traits in data which otherwise would remain hidden. We demonstrate the effectiveness of our method by applying it to two real-world datasets: traffic density in Barcelona and temporal measurements of air quality in Castile and León in Spain.

Index Terms—Visual analytics, networks, dimensionality reduction, data transformation

1 INTRODUCTION

Data visualization is extensively used to reveal patterns and structures in data. The display of high-dimensional datasets concerning point clouds with a high number of attributes continues to be a relevant research field due to the wide range of applications. The choice of an informative visualization technique depends not only on the characteristics of the data but also on the tasks to be performed. For example, a visualization to analyze the evolution of a high-dimensional time series requires a different approach than projecting a document corpus. While both aim to represent the data in a limited number of dimensions, the first emphasizes the progressive and continuous changes that occur in time and the second aims to find differences between groups of documents.

Dimensionality reduction techniques allow for embedding high-dimensional data into a plot with two or three axes. These solutions provide a visual scalability advantage over classical scatterplot matrices and parallel coordinates [1]. However, the low-dimensional transformations rely on assumptions and parameterizations which can compromise part of the original information. The most recent methods such t-SNE [2] or UMAP [3] are effective in identifying similar elements and projecting them separated from other groups. These projections favor the preservation of the closest neighbors rather than preserving all distances. Although global relationships are not omitted entirely between the points, the projections in the lower space can cause an incomplete mapping of the dataset [4].

On the other hand, Topological Data Analysis (TDA) aims to deduce and recognize geometric structures from underlying data by means of connecting elements in a graph. The combination of a scalar function with the original source allows exploration of data from different perspectives highlighting information which otherwise would be hidden. Unlike dimension reduction techniques the reconstruction of topology may not be faithful to the original data geometry, but does preserve the continuity between data shapes. Although TDA has demonstrated remarkable results in specialized studies [5], [6], [7], it relies on data summaries (e.g., clusters) instead of individual data points and therefore limits the resolution of the exploration phase. In addition, the cornerstone of TDA is clustering and appropriate lens which precludes hypothesis-free data exploration.

In this paper we present STAD, a parameter-free dimensionality reduction method which transforms the high-dimensional data into a graph highlighting the underlying structure. The projection into a graph provides a higher degree of flexibility to represent the interdependencies between points than coordinate mapping techniques (Fig. 1). Furthermore, STAD allows for incorporating additional functions which can intensify the specific signals adding new perspectives to the exploratory analysis. Additionally, STAD networks generate a representation of data without aggregation, i.e., STAD encodes the original data points as vertices in the graph which provides a high resolution of the data.

This paper is organized as follows. In Section 2 we give an overview of related work in the detection of structure in data through dimension reduction and exploratory techniques using graph representations. Section 3 describes the
proposed methodology, followed by Section 4, in which we present two case studies applying STAD. Section 5 includes an evaluation of STAD in comparison to other dimensionality reduction techniques. The discussion is presented in Section 6 and, finally, Section 7 covers conclusions and possible directions for future work.

2 RELATED WORK

The exploration of high-dimensional data has been presented in different areas of research in information visualization, data mining, and machine learning [9]. We review the related work in these areas below, more specifically the topics of dimension reduction, visualization techniques and TDA.

2.1 Dimension Reduction and Embeddings

Dimensionality reduction techniques transform a high-dimensional space to a lower-dimensional one. Considering as input the $N^2$ data points of a pairwise distance matrix, the methods to visualize the global structure falls in the multidimensional scaling (MDS) class [10]. Torgerson Scaling [11], a particular case of PCA, finds a linear and orthogonal transformation of data revealing the most informative view without modifying the local and global relationship between elements. More versatile approaches are non-linear metric MDS (NMDS) [12] and Sammon mapping [13], which overcome the linearity limitation introducing an iterative approach to match the distances between the original and projected space by minimizing the error between both matrices. The difference between Sammon mapping and non-linear MDS is that Sammon normalizes the original distance to emphasize small differences. The iterative MDS approach has been the basis for other models with improved versions of the loss functions [14].

The Isomap algorithm [15] is also based on the iterative MDS model, but using geodesic distances instead of Euclidean distance. The algorithm defines a neighboring region based on a parameter $\varepsilon$ given by the user. Once the neighbors are defined, the low-dimensional embedding is generated in a similar fashion to iterative MDS. This method eliminates the need for estimating distances between widely separated elements.

The underlying idea of associating the pairwise distances between the original space and a projected space is also employed in the STAD method. The difference with MDS techniques is that the projection takes place into a graph and more precisely in the path described by nodes and edges. The change in spatialization from a fixed coordinate system to a free-dimensional space provides a more flexible visual technique to represent the information as compared in Fig. 1. Instead of mapping the position in a lower space, the STAD graph aims to visualize the relationships between the data elements.

Beyond MDS techniques, more recent dimensionality reduction techniques such as t-SNE [2], LargeVis [16] and UMAP [3] improve the projection onto low-dimensional space by intensifying the detection of nearest neighbors. UMAP is able to organize these local signals more coherently according to original relationships between points or collection of points in the high-dimensional space, allowing global interpretations closer to MDS projections. These techniques assume an intrinsic probability distribution which smooths the projection in the low-dimensional space improving the detection of local patterns distorting the global one. However, they tend to increase the division among neighborhoods, which are beneficial for identifying local clusters but contrary to perceiving the relationships between them, and identifying global trends or continuous patterns [4].

2.2 Exploration of Data Through Network Structures

A number of earlier research projects used the network metaphor to facilitate the understanding of multidimensional data (e.g., [17], [18], [19], [20]). All these techniques depend on a parameter which determines the elements’ connectivity. The exploratory system presented by Jánick et al. [20] employs the the Minimum Spanning Tree (MST) to define the minimal structure of the data. Additional edges are added to establish a more consistent data structure using a force-directed graph layout. STAD uses the same concept of adding edges on top of a MST to draw the data shape but the number of edges are automatically selected through a minimization process. The structures presented in STAD generate a more accurate representation of the original
high-dimensional space by providing not only the structural shape but also an intuition of the density through the interconnection of nodes in a region of the graph.

Topology studies the global structure of a dataset from a geometrical perspective providing an informative summary. Topological Data Analysis is the general term used for a collection of particular methods to analyze high-dimensional datasets [21]. Graph representations are commonly used to illustrate the underlying structure of data, but nodes are aggregations of points rather than individual elements. Under this umbrella, data skeletonization is an important shape descriptor from a disconnected point cloud [22], [23], [24]. The selection of a proper skeleton is defined by the representation which shows the most persistent features. The stability of topological features is visualized in a so-called barcode [25] and analyzed to identify suitable parameterization [26], [27]. Other TDA methods rely on scalar functions to guide the summarization of high-dimensional data such as Morse-Smale Complexes (MSC) [28], [29], [30], Reeb graphs [31] and the Mapper algorithm introduced by Singh et al. [32]. While these functions are defined to determine the continuous space of a manifold, the functions in STAD modify the projections of distances by controlling the connections between nodes. However, since the same functions can be applied in both structures, they can be similar. In addition, the evaluation criteria in STAD rely on the association between the graph representation and the original space which differs from the geometrically persistent implications of TDA.

3 Methodology

Network visualization representations are projections of data expressed independently from a coordinate system; the visual structure connects elements according to their relationships and not their location. We extend the concept to represent the similarity (distance) between two nodes as the path described by the edges in the network. Once a similarity metric is chosen, a distance matrix $D_X$ containing the pairwise distances between all elements can be defined. The distance matrix can be considered a complete weighted graph $G_X$, where the indices of the matrix represent the vertices of the graph and the edge weights the distance between any two elements.

STAD proposes a new method to generate the structure of data by transforming the distance matrix into a graph. This method converts the complete graph $G_X$ into a non-complete unit-distance graph $U$ (i.e., all edges in the network have the same length of 1) where the distance between datapoints is reflected in the length of the shortest path between them. That is, the distance between two distant points is built from the neighborhoods of other nodes. A STAD network forms a single connected component, and it ensures the path for any combination of vertices exists. The information presented in the STAD networks is an approximation of the original complete weighted graph due to discretizing the distances in unitary segments. The number of edges in the unit-distance graph controls the shape of the data, and a final graph can vary from the minimum spanning tree to the complete graph. The STAD procedure selects the number of edges automatically by maximizing the correlation between the weighted distance matrix $D_X$ and the unit-distance graph.

In Section 3.1, we describe the details of the STAD algorithm and illustrate the steps with a simulated example. In Section 3.2, we present an extension of STAD to amplify signals in data through the addition of filters.

3.1 STAD Base Algorithm

The STAD algorithm can be split in eight sequential steps (Fig. 2): create the distance matrix, build an MST from the complete weighted graph, convert the MST to the unit-distance-graph, add edges to the unit-distance graph, evaluate the objective function, visualize the relationship between correlation and the number of edges, identify the optimal number of edges automatically and create a node-link diagram of the final network.

3.1.1 Create Distance Matrix in Original High-Dimensional Space

Let $X$ be a space with $n$ elements and $m$ dimensions in $\mathbb{R}^m$, and a metric exists which determines all pairwise distances $d_{ij}$ with $1 \leq i, j \leq n$. The distance matrix $D_X$ is the squared matrix with size $n \times n$ containing all the distances $d_{ij}$. This distance matrix $D_X$ is the only required element to generate a STAD network. From our perspective, the matrix $D_X$ is understood as an undirected, weighted and fully connected graph $G_X$ with $n$ vertices and $\frac{d_{ij}^2}{d_{ij}}$ edges where each edge $e_{ij}$ has a weight of value $d_{ij}$. Fig. 2a illustrates the distance matrix creation from a point cloud and the representation of the fully connected graph. The similarity between each pair of elements is projected as edges in the graph. Notice that edges in $G_X$ are undirected due to the symmetric property of the matrix and the diagonal elements are omitted in the graph.

3.1.2 Build MST From Complete Weighted Graph

Next, a minimum spanning tree (MST) is computed for $G_X$. The MST is a subset of $n - 1$ edges that connects all vertices without loops and with a minimal total sum of edge weights. Given this structure, the distance can be calculated between any two vertices as the length of the path connecting the two. Fig. 2b shows the MST network for the complete graph $G_X$. Note that the MST may not be unique and alternative combinations of edges can produce the same result.

3.1.3 Convert MST to Unit-Distance Graph

The MST is the first unit-distance graph $U_0$ considered in STAD (Fig. 2c) and the addition of edges will improve the association between $U$ and $G_X$. By transforming the complete graph $G_X$ into a unit-distance graph $U$, we reduce the graph dimension of the original into a two-dimensional graph formalized by Erdős [33]. Removing the edge weights and therefore converting the graph into a unit-distance graph where all edges have the same weight is necessary for calculating the subsequent STAD networks.

3.1.4 Add Edges into Unit-Distance Graph

The addition of new edges to the unit-distance graph $U$ produces a new graph $U_i$, where $i$ is the number of edges included in addition to the MST.
First, edges are sorted by weight to define the order in which they are added. For instance, \( e_{i-1} < e_i \) means that the weight of \( e_{i-1} \) is smaller than that of \( e_i \), or that the data points referred to in \( e_{i-1} \) are closer together in the original space than those referred to in \( e_i \). Then the edges are added into the network \( U_i \) following a cumulative process so that if \( e_1 < e_2 < e_3 < \cdots < e_{n-1} < e_n \) then \( U_1 \subseteq U_2 \subseteq U_3 \subseteq \cdots \subseteq U_{n-1} \subseteq U_n \). When \( U_i \) is the unit-distance graph with the sequence of edges \( e_1, e_2, e_3, \ldots, e_i \) from \( G_X \). Although edges are sorted and added into the unit-distance graph by their weight, all edges in \( U \) itself are unweighted.

3.1.5 Evaluate the Objective Function: Calculate Correlation Between Distances in Original Space and Those in Graph Space

From the unit-distance graph \( U_i \), the computation of the shortest path for every pair of vertices produces a squared matrix \( D_{U_i} \), with size \( n \times n \) comparable to the distance matrix \( D_X \) for the original space \( X \). The Pearson correlation is used to measure the agreement between \( D_{U_i} \) and \( D_X \). Contrary to the statistics absolute error, correlation is invariant under different scalings and takes a known range within -1 and 1 [34].

3.1.6 Optional: Visualize the Relationship Between Correlation and the Number of Edges

The evaluation of the correlation for consecutive values of \( i \) describes a quasi-convex function. The influence of an edge addition at the beginning of the sequence (i.e., at low values of \( i \)) has a large effect on the correlation with distance matrix \( D_U \). The evaluation of correlation at this stage may fluctuate when \( U \) contains few edges but describes a convex curve when the amount of edges is big enough. The number of edges needed to reach maximum correlation depends on the size and nature of the data. Intuitively, the association between \( D_X \) and \( D_U \) is related to the concept of persistence of clustering solutions [35], i.e., if the shape is persistent along the edges, the computed correlation will be consistently similar.
3.1.7 Identify the Optimal Number of Edges Automatically

STAD uses simulated annealing (SA) [36] to estimate the optimal \( U_i \), which maximizes the correlation between the distances from the STAD graph and the original distance matrix, and provides a representative data projection of \( X \). SA approximates the combination of links which maximize the correlation between \( D_{U_i} \) and \( D_X \). This heuristic is a stochastic process and estimates the global optimum by exploring the discrete space of edges (Fig. 2g). The SA candidate generator reduces failures on non-convex regions produced by the correlation, mainly when the graph is composed of only a few edges. Note that as the resulting network is ultimately explored visually and structural features are kept along networks with a similar number of edges (persistence of structural features), the difference between a global maximum and an approximation thereof does not imply noticeable deviations in STAD. The start and end of these ranges are only identifiable after the evaluation of all links. Fortunately, these do not all have to be computed, as the most important characteristics are also the most easily detectable.

The time complexity of calculating \( D_{U_i} \) is \( O(|V| + |E|) \) as described in the Breadth-first search algorithm definition in [37]; where \(|V|\) is the number of vertices and \(|E|\) is the number of edges in the graph. Although the number of vertices is fixed, the number of edges evaluated at iteration \( i \) influences the running time and varies between \( O(|V - 1|) \) when the MST is evaluated and \( O(|V|^2) \) when the graph is complete.

3.1.8 Create Node-Link Diagram of Final Network

Networks require a graphical convention to be visualized. Generally, they are drawn as node-links representations projected in the two-dimensional plane. Although the STAD methodology generates an unweighted graph (unit-distance graph), we include the distances from the original metric \( D_X \) in the final node-link diagram as this improves the visual representation. Fig. 2h shows the resulting network including the distances as weights in the edges. Note that the STAD graph is independent of the graph drawing algorithm used, the focus is on the identification of signals described by the connections of elements rather than the coordinates of nodes. However, graph drawings which minimize the number of crossings and place together small edge weight are appropriate to detect data structures, e.g., ForceAtlas2 [38] and Kamada-Kawai [39].

3.2 Filters in STAD

The transformation of datapoints and weights from a complete graph into a unit-distance graph representation can hide part of the information, and we can expect that not all patterns in data will be revealed in a STAD graph. In particular, prior knowledge might reveal that for certain applications, specific datapoints should be pulled apart even if they are located close together when considering the complete high-dimensional space. We propose an extension of the STAD base algorithm to highlight other signals in the data by the inclusion of functions that act as filters on data projections.

Filters are an ordered set of values associated with an indexed sequence of natural numbers. They provide limits and thus a context to an arbitrary metric space. Filters can be defined from derived dimensions through statistical functions, subsets of dimensions or external data. Real sequences may be discretized by defining equidistant intervals based on scale or density. The addition of filters focuses on the exploration of data allowing, for example, to integrate domain knowledge. Formally, filters are defined as a space \( Z \) with \( n \) elements and \( p \) dimensions in \( \mathbb{R}^p \) where a mapping exists between \( X \) and \( Z \), i.e., there is a function \( f : X \to Z \). Filter functions theory is also present in topological methods [40] and the same filters used in TDA can be applied in STAD, providing in some cases similar shapes. However, although both share similarities conceptually, the fundamentals are different. While STAD aims to accentuate particular traits in the projection of a non-continuous set of points through filter functions, in TDA the filters are the basis of the projection itself and are used as an instrument to generate the manifold.

The inclusion of a filter replaces the first two steps of the STAD base algorithm (Sections 3.1.1 and 3.1.2) by a new approach composed by three steps: define the filter (Fig. 3a), create a reduced distance matrix based on this filter (Fig. 3b), and build a MST from the semi-complete weighted graph (Fig. 3c).
3.2.1 Define the Filter Function
Filter definition depends on data characteristics and the purpose of the analysis. As with other topological techniques, density estimates, centrality functions, orthogonal coordinates, a subset of dimensions and intermediate algorithmic results [41], [42] are also supported in STAD. The inclusion of filters aims to enrich data exploration through explicit prior knowledge [43] rather than hypothesis-free research. In practice, subsets of dimensions or external data tend to be most interpretable.

Filters in STAD are understood as a linear space where data follows a sequential order. However, the nature of the data included in a filter Z can be diverse, and both the filter definition and interpretation must be adapted to it. For example, cyclical patterns in temporal data are common such as the day of the week or the month in a year. The last element of this cyclical pattern is as far from the previous as it is from the following although the sequential labeling does not reflect this, i.e., Sunday (day 7) is close to Monday (day 1) as is December (month 12) to January (month 1). In these cases, filters Z must be represented in a polar space where the repetitive pattern is preserved [44].

Filters are mostly defined as one-dimensional or two-dimensional. Higher dimensionality although possible tends to over-restrict the space. When filters are real, a discretization process is required to transform them into a natural sequence of indices. In this paper, we present the real filter transformation by specifying r as the number of intervals to divide each dimension in. Fig. 3a illustrates the transformation of a real filter into a natural sequence. The effect of variations in the value of r during the transformation influences the final network. The implications are discussed in Section 3.3. The value r can take independent values for each dimension when the filter dimension is greater than one, but the intervals must allow forming a single connected component network as STAD output. Empty intervals in a one-dimensional filter are omitted and the adjacency of the intervals is considered to the closest non-empty range. In filters of higher dimensionality, empty intervals are evaluated together with their neighbors defining a consistent grid. The algorithm to generate a consistent grid in STAD is provided as supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2020.2995465.

3.2.2 Create Reduced Distance Matrix Based on Filter
The inclusion of a filter Z establishes limits in the metric. We can use these boundaries to introduce the effect of the filter in STAD by reducing the distance matrix $D_X$ and in consequence the complete graph $G_X$. We define three types of possible connections between datapoints (Fig. 3a):

- **Intra-edges** are all connections $e_{ij}$ where $i$ and $j$ belong to the same index.
- **Inter-adjacents** are all connections $e_{ij}$ where $i$ and $j$ belong to adjacent indices.
- **Inter-nonadjacents** are all connections $e_{ij}$ where $i$ or $j$ belong to different, non-adjacent indices.

Based on these definitions, the distance matrix $D_X$ is reduced to $D_X^*$ by removing all non-adjacent connections (Fig. 3b). The STAD process uses the distance matrix $D_X^*$ as input in the estimation of the filter. The derived graph from the distances becomes a semi-complete weighted graph $G_X^*$ where only links within and between adjacent intervals are considered. The reduction of connections draws networks based on the structure of the filter highlighting properties of data such as the temporality of time-series or abnormality level of a centrality measure. Additionally, the performance of STAD with filters improves due to the smaller size of the distance matrix to be evaluated.

3.2.3 Build MST From Semi-Complete Weighted Graph
From $G_X^*$ the MST can be computed as described in Section 3.1.2. However, one might want to ensure certain datapoints to be close together based on specific domain knowledge, even if they are further apart in high-dimensional space (or vice versa). In case the specific domain knowledge is expressed in a particular dimension, this would mean that the datapoints are far apart when considering all dimensions, but close together in the dimension under consideration.

Although the classical MST provides valid results in STAD with filters, we propose a version of the MST which better preserves the filter structure by prioritizing intra-edges in the process. Artificial connections (i.e., connections made as an artifact of splitting the data along the filters) are detected through community detection and re-evaluated globally. This process is split into these three steps:

1) The MST$_{intra}$ is created first inside of each index (intra-edge connections). Inter-adjacent connections are added after the MST$_{intra}$ computation to define a single connected component (Fig. 3c left).

2) The intra-edge connections from MST$_{intra}$ are evaluated through community detection using the original distances as weights. We implemented the random walk methodology Walktrap [45] due to its adaptability to short sequences. This step aims to detect distant points in high-dimensional space that were connected inside of each index. A sensitive configuration of community detection is desired to detect the different signals of data, as false negative divisions are automatically corrected in the following steps. All intra-edges falling in the same community are preserved and fixed. Remaining edges, i.e., discrepant intra-edge and inter-adjacent edges are omitted and re-evaluated in the following step (Fig. 3c center).

3) Edges from the previous steps are preserved and act as a base, and additional edges are added until a single connected component is created (Fig. 3c right).

3.3 STAD Network Interpretation
STAD networks generate shapes which provide both global and local intuition of a data structure. Local signals can refer to clusters, i.e., a homogeneous group of data points according to their similarity, but also broader meanings, for instance, a set of points with gradual dissimilarity (which presents itself as a flare). The graph density provides a notion of data distribution; homogeneous elements appear highly interconnected in the graph and dissimilar elements appear in non-adjacent sections of the graph. The visual edge length
of STAD graphs indicates the similarity between the two vertices (see Section 3.1.8). Specific graph layout algorithms such as force-directed layout will search for an equilibrium between these edge lengths and their optimization function, e.g., to minimize overlap of nodes and/or edges [46].

The inclusion of filters intensifies specific information contained in their dimensions. Fig. 4 exposes the effect of filters in a comparable setting where the same variable has been split in a different number of natural indices. According to the distance matrix detailed in Section 3.2.2, when the number of filter indices is two or smaller, no non-adjacent connections exist and therefore we obtain a STAD network identical to the filter-free approach (Fig. 4b). A higher number of indices produces a fine-grained representation of the filter definition but penalize the structural representation of points contained in the underlying dataset X. If the number of indices in the filter is equivalent to the number of elements in the dataset, the generated network is forced to connect the adjacent indices. In this case, the STAD network exposes the structure of the filter instead of the structure of data. Additional features of the graph (e.g., node color and/or size as used below) can aid in the interpretation of the data.

4 Case Studies

We applied STAD to two real-world datasets. We present results, derived insights extracted from the shapes and discuss choices on the filter selection. The visual analytics approach helps to discover non-evident patterns in data through the connection between the points describing data shapes as flares and loops.

4.1 Barcelona Traffic

We collected a dataset from the public repository Open Data BCN [47] which contains traffic activity in the city of Barcelona. The analysis was performed with all records from October 2017 until November 2018 (374 days). The dataset describes measurements of traffic density collected every five minutes in 534 locations of the city which is stored as an ordinal variable from one to six, one corresponding to freely moving traffic and six to stand-still. We explored the daily changes in the city by averaging the individual sections into a one-dimensional time series for each day aggregated by hour. Similarity between days was computed using euclidean distance to identify differences at identical timestamps. In this section, we will discuss two analyses: one without filter and one using a two-dimensional filter composed of the week number and the daily mean of the densest point in the city.

4.1.1 Filter-Free STAD Analysis of Barcelona Traffic

Filter-free STAD analysis results are shown in Fig. 5 where we identify three different patterns. In this example, nodes are colored by day of the week (blue shades correspond to workdays from Monday to Friday, orange refers to Saturday, and red to Sunday) and the size of the node indicates the mean of traffic density in that day. The most significant signals correspond to the difference in activity between weekdays (Fig. 5a) and weekends (Fig. 5c). The groups of workdays on the left are highly connected indicating the high similarity between these days. In the center of the graph (Fig. 5b), we find a subset of days between the largest group of workdays and weekends. This subset corresponds to lower activity days in the city; more specifically, they are workdays in the first week of January, Easter week and the month of August. These periods of the year traditionally correspond to the two days of the weekend. Saturdays are associated with holiday periods and are distinguishable from the rest of patterns in data. On the right of the graph, we recognize the weekend and official holidays. Inside this sub-network, we recognize two more groups which mostly correspond to the two days of the weekend. Saturdays are days with higher activity than Sundays as reflected on the node size of the figure. Official holidays behave like a typical Sunday; we highlight Christmas day as the day of the year with the smallest traffic activity, located at the extreme top-right. In contrast, there are some Sundays with higher traffic activity which have been related to some featured event. For example, we name the Political Prisoners Demonstration on April 15th, the 40th Zurich Barcelona Marathon on March 11th and the Final World Cup 2018 on July 15. These days are associated with a higher movement of people and the closing of some section of the city.

Although the connectivity of the network does not provide additional structural insights, the color of nodes helps to recognize weaker signals in the graph as well. More concretely, we can see that Fridays are particularly clustered. Digging deeper into the data we can identify a peak of activity between 14:00 and 16:00 on Fridays (see image in supplemental material, available online). The increased traffic is associated with departures leaving the city.
The global structure of the network presents coherent connectivity between the groups according to their traffic density: the group with highest traffic congestion i.e., typical workdays (Fig. 5a) connects to the group with workdays on holiday period (Fig. 5b) and this is linked to the weekends (Fig. 5c).

4.1.2 STAD Analysis of Barcelona Traffic Using Two-Dimensional Filter

We continue the analysis of Barcelona traffic by incorporating filter functions to identify additional signals. We applied STAD using a two-dimensional filter composed of the week number and the mean of the densest point in the city for each day. The resulting network in Fig. 6 maintains the three groups from the approach without filters (Fig. 5) but additional features are revealed. Two additional loops are present, one in the group of workdays and the other on weekends. Further investigation indicated that these structures correspond to renovation works [48] starting in May 2018 which resulted in the closing of a transversal avenue in the west of the city.

The visual gaps in the graph, for instance, between groups a and b, are created by public or bank holidays, which end up in the cluster of the weekend days (groups c-d and g) and generate this weaker connectivity between the indices of the temporal filter week number. Likewise, the gap between groups e and f is due to August present at the center of the graph. The circular pattern of traffic between years is reflected in the connectivity between groups f and a. In the weekends we can identify the same separation due to the renovation works (c-d versus h-i).

4.2 Air-Quality in Castile and León

We applied STAD on the air-quality dataset collected from the Castile and León initiative in Spain [49] to illustrate STAD as a visualization tool for the identification of patterns on high-dimensional time series. The dataset contains daily measurements of seven chemicals such as carbon monoxide (CO), nitrogen oxide (NO), nitrogen dioxide (NO2), ozone (O3), sulfur dioxide (SO2) and particulate matter 10 and 2.5 (PM10 and PM2.5). The measurements have been collected at different locations from January 1997 to June 2018. The data was aggregated by week due to the presence of missing values. The explored multidimensional time series contains 1,139 records with seven dimensions, and we computed the euclidean distance to evaluate the similarity between elements. The resulting STAD network graph is presented at Fig. 7. The structural shape generates an intrinsic separation of time identifying changes in the air-quality over the years. We recognize three dense groups of points which mainly correspond to different periods: 1997-2002 (Fig. 7a), 2003-2008 (Fig. 7b) and 2009-2018 (Fig. 7c). These visual splits identify relevant changes in the air-quality. The vertical position of nodes provides an intuition of seasonality, i.e., nodes on top of the network correspond to autumn-winter dates and nodes on the bottom to spring-summer. The coloring of nodes by seasonality is provided as supplemental material, available online.

To investigate seasonality signals further, we extend our exploration by incorporating the week number as the filter in STAD (Fig. 8). The network conserves the signals identified in Fig. 7 although they also reveal additional ones. For instance, between 1997 and 2002 (Fig. 8a) two groups are evident, corresponding to different seasons (summer-winter and spring-summer). In contrast, between 2009 and 2018...
The two different groups identified at (Fig. 8a) are mainly caused by the chemicals nitrogen oxide and dioxide, carbon monoxide, and sulfur dioxide. These measurements are higher during the autumn-winter and are related to the burning of fossil fuels [50]. In recent years, electric systems started to substitute the previous technologies [51]. This fact is visible in the period 2009 to 2018 (Fig. 8c) where the difference between seasons is less evident.

The gap between 2002 and 2003 at spring-summer (Fig. 8b) indicates the decrease of particulate matter PM10 and PM2.5, which is related to vehicle emissions [52]. This period corresponds to new vehicle restrictions [53].

- Different coloring of nodes may help reveal additional patterns in data (see images in supplemental material, available online). For example, ozone fluctuates according to the season period. During the spring-summer, ozone levels are higher due to variations in sunlight and UV radiation [54]. In addition, concentration of carbon monoxide, nitrogen, ozone, sulfur dioxide, and particulate matter decreases gradually over the years, reaching stability in 2010. This finding is associated with an improvement of air-quality [55].

5 Evaluation

The main novelty of STAD lies in the way distances are represented as a network, providing not only a two-dimensional mapping of data but also connections between nodes.
which reinforce the communicated information in the plot. A clear distinction needs to be made between the network as the underlying data structure and the visual representation of that network on paper or a computer screen. Unlike the explicit positioning of datapoints in a scatterplot, node locations can be modified for example so that overlapping of nodes and/or edges is reduced. At the same time, the links in the network do not change.

In this section, we present a quantitative and a qualitative evaluation. The quantitative comparison aims to verify if STAD networks adequately capture the distances between datapoints in the original multi-dimensional space, both within the graph structure and within the projection of that graph onto a 2D plane. With the qualitative assessment we collect advantages and disadvantages of STAD networks over other dimensionality reduction methods based on scatterplots.

5.1 Quantitative Evaluation
The quantitative evaluation covers the comparison from two angles: the STAD graph structure (as an abstract data type) and the 2D-projection of the STAD network as a node-link diagram. For the latter, different layouts place vertices according to different criteria. For example, the Kamada-Kawai layout aims to minimize the discrepancy between distances in the graph and projections using a stress function equivalent to the NMDS approach for graphs [39]. ForceAtlas2 leverages attractive and repulsive forces based on Barnes Hut simulations in order to obtain a layout where edge lengths are small while vertices are well-separated. This energy model implemented in ForceAtlas2 has an impact on the shape of the graph, generating clusters of nodes tighter, for example, than Kamada-Kawai [38].

We perform two main analyses in this section. First, the preservation of the global distance was measured using the Spearman rank correlation of the distances in the projected space and those in the original multidimensional space (Fig. 9) [56]. In other words, for every pair of points we compared their distance in the original space with that in the projected space. Second, local neighborhood preservation was determined by the proportion of neighbors preserved in the projection compared to the original space (Fig. 10) [57].

Preservation of global distance - The Shepard diagrams in Fig. 9 present the point-pair comparison of distances for three dimensionality reduction methods (Figs. 9a, 9b, and 9c), the distances in the underlying STAD graph (Fig. 9d), and those in two graph layouts (Figs. 9e and 9f). Note that graph layouts only consider node placement, and the visual influence of links cannot be measured quantitatively. Whereas the STAD
graph itself (Fig. 9d) only employs the connections between nodes, the layout algorithms in Figs. 9e and 9f may distort edge lengths.

Based on the Spearman rank correlation, NMDS conserves distances the best between the original and projected space ($\rho = 0.99$). This is closely followed by the STAD graph with a Spearman correlation of 0.98. The STAD graph tends to underestimate distant datapoints in the network structure. This effect is expected in STAD because the representation of distances depends on connections between other nodes and by definition the STAD network creates a single connected component. Although the placement of nodes in two dimensions using the Kamada-Kawai layout has a Spearman correlation lower than NMDS, the value is higher than other dimensionality reduction techniques (t-SNE and UMAP in Figs. 9b and 9c respectively). The t-SNE and UMAP methods emphasize the relation of datapoints with their closest neighbors over that with distant datapoints, although for this analysis their neighboring parameters were maximized in this evaluation to benefit the projection of global data structure. Nevertheless, this results in a penalization of global distances compared to other algorithms like NMD or STAD using a Kamada-Kawai layout. The UMAP example (Fig. 9c) clearly shows the impact of connecting the Shepard diagram, where two clouds of points can be recognized (groups I and II). Elements contained in group I represent distances within weekends and weekdays. However, group II indicates distances between day where one is in the weekend and the other is not. The results of STAD using a ForceAtlas2 layout generate a Shepard diagram (Fig. 9f) with a similar structure to UMAP. Although the Spearman correlation is lower than UMAP, the two groups (i.e., groups I and II) are equally identifiable. In addition, group III is more visible, corresponding to distances between workdays during the holiday period (Fig. 5b) and weekends (Fig. 5c).

Preservation of local neighborhood - In the second analysis, we measured the proportion of neighbors preserved at different neighborhood sizes: on average, how many of the $k$ nearest neighbors in the original space were also within the $k$ nearest neighbors after projection (Fig. 10). The abstract STAD graph preserves the local neighborhood almost perfectly for the first six closest neighbors. Indeed, defining nearest neighbors constitutes an essential part of the STAD methodology: the MST graph is the starting point for a process that iteratively adds more edges (Section 3.1.2).

The abstract graph structure is conceptually not the same as the node-link visual representation. The position of nodes in the node-link visual representation is limited to two or three dimensions to project all relationships like other dimension reduction methods do. In this example, ForceAtlas2 obtained slightly better results than Kamada-Kawai and the other three-dimensional projections included in the comparison. Even though the graph drawing algorithms aim to facilitate the legibility of networks rather than the preservation of distances at any scale, this result indicates that repulsion and attraction force systems tend to preserve local neighborhoods.

### 5.2 Qualitative Evaluation

The main claim of STAD is that network visualization is a flexible platform to display complex patterns in data. To assess the usefulness of the STAD approach, we performed interviews with eleven participants (participant A to K), including doctoral researchers, post-doctoral researchers, and two professors in the area of data analysis. None had experience with STAD before the interview. We conducted the interviews using the Barcelona traffic case, comparing the visualization of three dimensionality reduction methods (Figs. 11a, 11b, and 11c) and STAD using two different graph layouts (Kamada-Kawai layout as shown in Fig. 11d and ForceAtlas2 as illustrated in Fig. 5). A brief introduction was given, explaining the type of data and analysis conducted. In the first stage, all node-link diagrams were monochrome, and the graph structure itself was the only source of information. We asked participants to think aloud about plots, highlight relevant regions that evoke clusters, trends, and outliers. In a second stage, the colored plots as in Fig. 5 were presented to confirm or modify their previous findings. To conclude the interview, we asked their personal view about networks as visualization metaphor for high-dimensional data.

During the process of analyzing the monochrome plots, all participants identified at least the two big groups of days corresponding mostly to workdays and weekends (groups 1 and 3 in Fig. 11). Five participants (A, C, H, I, and J) described explicit uncertainty identifying groups in plots. This uncertainty was mentioned during the exploration of t-SNE or UMAP when participants were interpreting the group in the middle (i.e., lower activity traffic during the holiday period present in group 2 of plots in Fig. 11). They doubted if this set was a different signal or random noise in the sample. On the other hand, three participants (E-G) identified the holiday group as an independent signal from workdays and weekends group in t-SNE, UMAP, or NMDS plots. Four participants (A, C, E, and I) identified potential outliers. In contrast, all participants identified at least the three groups of patterns (workdays, weekends, and holiday periods) in the STAD plots (Fig. 11d), but none described or mentioned outliers.

In the second part of the interview we used the same plots, but with nodes colored by day of the week. Participants were
asked to review their previous selections, find similar patterns, and open a discussion about STAD graphs. Ten of the eleven participants (all except participant B) found the exploration easier using STAD graphs than using scatterplots, as the former generated more interpretable results to understand the data. The most common reasoning behind this was the presence of links which facilitate the identification of groups. For example, participant I said: ‘edges suggest the groups of data’, participant J ‘STAD imposes clusters, ..., links provide meaningful information’, and participant D ‘connections make the interpretation of traffic easier’. Despite the difficulties of participant B, he recognized that STAD networks provide insights not present in scatterplots. However, links make it harder to identify densities in data.

Based on the findings from these interviews, we can deduce that networks produce more defined structures than scatterplots, although the same information can be recognized in both visuals. In addition, similarity between data elements is represented not only by node positions but also by the number of connections that a node receives. At the same time, as STAD networks consist of a single connected component, single outliers are less visible than with other methods.

6 Discussion

In this section, we discuss some limitations of the STAD methodology, their possible solutions and open challenges that remain to be addressed.

6.1 Scalability

While STAD analysis of some datasets results in sparse networks with easily interpretable structures, other datasets end up represented in more complicated networks. As the algorithm works at the level of the individual datapoints, the analysis of large datasets comes at a significant computational cost. In addition, drawing of a resultant large network also becomes cumbersome.

Based on the Shepard diagrams, STAD gives a similar accuracy as NMDS, which computes the pairwise distances between all points in over several iterations. Therefore, running times of these two methods is comparable. However, the computation of shortest paths is slower than euclidean distance which penalizes STAD over NMDS as it comes to speed. For instance, construction of the projection of the largest dataset in this paper (air-quality dataset) takes 1.2 minutes for STAD and 1.0 minutes for NMDS. These results are still far from local distance-preserving methods such as t-SNE and UMAP, which takes 16 seconds for defining the projection in two dimensions. This comparison was performed in R with a single-thread and ran on an Apple laptop MacBook Pro (dual-core, Mid 2014).

6.1.1 Computational Scalability

In STAD, the recursive computation of distances in the unit-distance graph comprises the main bottleneck of network estimation. A possible approach to alleviate this issue is to work with a smaller sample of the initial dataset. Preliminary tests have indicated that such smaller dataset retains the same visual structure as the full-size dataset, while not suffering from the high computation cost. The addition of edges upon the MST is a cumulative process (Section 3.1.4), i.e., if an edge \( e_i \) with weight \( w_i \) - with larger weight meaning larger dissimilarity - is added into the network all edges with smaller weight are part of the network \( U_i \). When the algorithm determines the optimal network, it finds an edge-weight threshold which establishes the resulting number of connections. This optimum can be calculated on a subsample of the full-size dataset. Multiple iterations on a down-sampled dataset can be performed in a parallelized setting providing a more robust threshold estimation. Other (faster and more advanced) approaches to calculating the intermediate STAD distance matrices are currently under investigation.
6.1.2 Visual Scalability
When large networks are considered, the number of links might become an issue for visualization, resulting in the dreaded “hairball”. Current approaches on graph layouts [38] could manage up to a million nodes if the network is sparse [46], [58]. Consequently, additional transformations such as aggregation to reduce the number of nodes and/or edges are still needed. A possible solution can be found in community detection pipelines [59], such as MCLEAN [18], that simplify this visual representation.

6.2 Addition of Edges
As mentioned in Section 6.1.1, the addition of edges from the MST follows a sequential and cumulative procedure based on their distances. The incremental approach may cause edge redundancy and contribute to increasing clutter in the plot. A non-additive and refined procedure such as an evolutionary algorithms [60] might reduce the number of links conserving the association with the original distance matrix $D_X$. Nevertheless, the use of evolutionary algorithm would also penalize the performance on the network estimation due to the assessment of crossovers in every iteration. Moreover, elimination of unnecessary edges does not change the structure of the resultant network, and such approach does not guarantee a benefit in making new features recognizable.

6.3 Stability and Reproducibility
The optimal number of links is dictated by the association between $D_X$ and $D_Y$. The SA algorithm estimates an optimum through a stochastic procedure which can generate different results in each iteration. However, the correlation curve describes a soft convexity with a quasi-constant function around the maximum.

Filters can bring additional signals in the underlying structure of data to the foreground. In extreme cases with very small bins, these might however fragmentize the original structure, although these changes have demonstrated to be reasonably robust (Fig. 4).

6.4 Evaluation
The evaluation of a STAD network comprises not only the comparison of metrics but also the qualitative assessment of recognized patterns as presented in Section 5. The STAD approach is designed to maximize the relationships between networks and distance matrices according to Pearson’s correlation. Fig. 1 provides an example of the intuition generated by STAD networks using a three-dimensional point-cloud of a horse. The continuous signals defined by the four legs of the animal are identifiable in STAD networks (Figs. 1d and 1e) in contrast to the overlapping produced by the NMDS mapping (Fig. 1a). Local preserving methods such as t-SNE (Fig. 1b) and UMAP (Fig. 1c) can still recognize several of the data patterns, but the nature of these techniques do not ensure a connected structure.

Although static representations of the STAD network as a node-link diagram are not necessarily better or worse than other approaches (see Fig. 10 and Table 1), the added value of the STAD network lies in its underlying graph structure rather than the positioning of the nodes in a 2D plane. Indeed, adding interaction to these representations where the user can click and drag nodes to other positions is an important method for better understanding the structure of the network as some nodes will be moving along as they are closely linked to the node that is dragged, while others are not.

Therefore, it is important to evaluate STAD projections both qualitatively (i.e., the qualitative identification of signals in data) and quantitatively (i.e., the representability of data patterns).

6.5 Other Related Techniques
Alternative methodologies can reveal equivalent signals in data to STAD, especially in the filter-free approach (see Fig. 11). Nevertheless, one of the added values of STAD lies in the preservation of data structures in lower dimensions. By encoding the distance in edges, the uncertainty of signals is mitigated because data elements are explicitly connected, revealing their shape. These definite structures are especially useful during the exploratory phase of an unknown sample.

Even though the construction of networks from non-relational data can be achieved using simpler methods such as k-nearest neighbor graphs, the data structure using these methods are poorly perceived, as shown in Fig. 11f. The non-uniform connectivity in STAD enables the identification of data with unequal densities, which is critical to recognizing different types of patterns. In addition, determining the best number of links in k-nearest neighbor graphs is still an open challenge in contrast to the parameter-free STAD methodology.

The recognized shape in STAD must be seen as a data visualization result rather than a topological one that could be obtained through methods such as the Mapper algorithm (see Fig. 11). The results of the Mapper algorithm are summaries of data that depend on function (lenses). Although these lenses share similarities with STAD, filter functions in STAD are used to intensify the signals contained in the function but without changing the original projected data. For this reason, STAD networks can be visually compared with and without lenses and discernible traceability between plots can be recognized, i.e., a similar global structure with different local patterns.

7 Conclusion and Future Work
With STAD, we propose a parameter-free methodology to visualize the structure of high-dimensional datasets as networks, allowing for the identification of signals by means of shapes as flares and loops. The network metaphor has been demonstrated to provide better clarity to visualize data than
other methods represented as scatterplots. Edges in the graph correspond to similarity between datapoints; therefore, similarities between individual datapoints are used to encode the higher-level patterns in the resultant graph and the resulting visualization is a compressed projection of the distance matrix into a free-scale space. In addition, integrating filters adds an additional perspective to the exploratory analysis.

An R implementation is available at https://github.com/vda-lab/stad; Python and Clojure implementations are under development. Results presented in this paper have been generated with this R implementation. The final graphs included in Section 4 were enhanced through Gephi [61].

Future work includes improving the efficiency of computational methods by retaining the information from previous iterations and approximating the shortest path [62]. In addition, we also aim to devise novel visual approaches to compare and interpret networks structures in an integrated environment.

ACKNOWLEDGMENTS

The authors would like to thank Danai Kafetzaki for valuable feedback and proofreading. This work was supported in part by the IWT/SBO 150056 project “ACQuirin g CrUical Medical information Using LAnguage TECnology” (ACCUMULATE), and by the Flanders AI Impulse Program (“Onderzoeksprogramma Artficiële Intelligente (AI) Vlaanderen”).

REFERENCES

Daniel Alcaide received the master’s degree in statistics and operational research from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2012, and the master’s degree in visual analytics and big data from the Universidad Internacional de la Rioja, Logrono, Spain, in 2015. He is currently working toward the PhD degree with the Department of Electrical Engineering, KU Leuven, Leuven, Belgium. He conducts research in interactive visualisation and data analysis to bring insight into large and complex data. He worked as a statistician in various areas related to clinical development in Barcelona where he learned about the power of visual representation to communicate information.

Jan Aerts received the PhD degree in genetics and genomics. He turned to the field of visual analytics in 2010. He is currently a part-time associate professor with KU Leuven as well as full professor with Hasselt University where he also heads the Data Science Institute. His research focus lies in the role of contextualization in the exploration of large and complex datasets. This includes research in visual design, interaction design and (human and computational) scalability towards dimensionality reduction and clustering. He has been on the organising committees of several conferences (including BioVis and Beyond The Genome), and has chaired visualization-related sessions at conferences including VIZBI, the Bioinformatics Open Source Conference BOSC and EuroVis/VML.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdrl.