Improving the Visual Analysis of High-dimensional Datasets Using Quality Measures

Georgia Albuquerque*
TU Braunschweig
Germany

Martin Eisemann†
TU Braunschweig
Germany

Dirk J. Lehmann‡
University of Magdeburg
Germany

Holger Theisel§
University of Magdeburg
Germany

Marcus Magnor¶
TU Braunschweig
Germany

ABSTRACT

Modern visualization methods are needed to cope with very high-dimensional data. Efficient visual analytical techniques are required to extract the information content in these data. The large number of possible projections for each method, which usually grow quadratically or even exponentially with the number of dimensions, urges the necessity to employ automatic reduction techniques, automatic sorting or selecting the projections, based on their information-bearing content. Different quality measures have been successfully applied for several specified user tasks and established visualization techniques, like *Scatterplots*, *Scatterplot Matrices* or *Parallel Coordinates*. Many other popular visualization techniques exist, but due to the structural differences, the measures are not directly applicable to them and new approaches are needed. In this paper we propose new quality measures for three popular visualization methods: *Radviz*, *Pixel-Oriented Displays* and *Table Lenses*. Our experiments show that these measures efficiently guide the visual analysis task.

Index Terms: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—; I.3.3 [Computer Graphics]: Picture/Image Generation;

1 INTRODUCTION

Although diverse visualization methods support the exploration of high-dimensional datasets, the visual analysis of such data is still a challenging task. The visualization and analysis of multivariate datasets typically involves mapping the data to lower dimensional embeddings, which is the case for Scatterplots or pixel-oriented methods, or determining a placement of the dimensions in multivariate visualizations, as in Parallel Coordinates [13] or Radviz [9]. This approach may generate several hundreds or thousands possible projections for high-dimensional datasets and the visual analysis may quickly become an overwhelming task. An alternative solution to this problem is to use a handful of quality measures to automatically select information-bearing projections of the data.

Quality measures are generally based on a specific user task, and may be used as a starting point in the visual analysis of multivariate data. Since the pioneering *Projection Pursuit* approach [6, 12], which searches for low-dimensional projections that expose interesting structures in the data, was presented, diverse indices and quality measures have been presented. Recently, innovative approaches using quality measures for specific visualization methods have been proposed for Scatterplots and Parallel Coordinates in particular[22, 23], but many other popular visualization methods still have to be explored in this sense.

In this paper we extend the existing set of quality measures and introduce new techniques for three other popular visualization methods: Radviz, Pixel-Oriented Displays and Table Lens. Our motivation to develop new quality measures for other visualization types is that the visual analysis is usually performed on different visualizations simultaneously, and that dimensions selected by a quality measure for a specific visualization method do not necessarily produce good projections for other visualization methods. We believe that the visual analysis benefits from automatically selecting the potentially insightful candidate projections in different visualization techniques. The relevance of the projections is determined based on the structures present in the visualization image that may indicate trends in the data, like clusters, outliers or correlations. Our three main contributions are: a dimension reordering algorithm to improve the visualization potential of the Radviz; a quality measure to rank Pixel-Oriented Displays; a measure to improve the usability of the Table Lens method. We tested our approaches on class-based and non-class-based datasets; the results show that our measures successfully support the user in the search for insightful visualizations and potentially speed up the visual exploration task.

2 BACKGROUND

Radviz, Pixel-Oriented Displays and Table Lens are well-known and accepted visualization methods for high-dimensional datasets. We propose the use of quality measures to automatically support the visual analysis task using these three methods. Such measures can be exploited to select information-bearing projections that may be used as a starting point in the visual analysis.

To exhaustively analyze a dataset using low-dimensional projections, Asimov presented the *Grand Tour* [3] that supplies the user with a complete overview of the data by generating sequences of orthogonal two-dimensional projections. The problem with this approach is that an extensive exploration of a high-dimensional dataset is effortful and time consuming, therefore quality measures are needed to select only the good views of a dataset.

As aforementioned, since the *Projection Pursuit* approach was introduced [6, 12], diverse indices and quality measures have been proposed. The goal of *Projection Pursuit* was to search for low-dimensional representations of a high-dimensional dataset where structures of the data could be observed. Later on, the Scagnostics method was proposed by Tukey *et al.* [24] to analyze high-dimensional datasets and Wilkinson presented more detailed graph-theoretic measures [27] for computing the Scagnostics indices for Scatterplots. Recently, diverse extensions to such indices, henceforth termed quality measures, were proposed to rank low-dimensional projections of the data based on specific visualization methods. In [22] class consistency quality measures were introduced to rank Scatterplots based on the class information of class-based datasets. In [23] quality measures for Scatterplots and Par-

*e-mail:georgia@cg.cs.tu-bs.de
†e-mail:eisemann@cg.cs.tu-bs.de
‡e-mail:dirk@isg.cs.uni-magdeburg.de
§e-mail:theisel@isg.cs.uni-magdeburg.de
¶e-mail:magnor@cg.cs.tu-bs.de

IEEE Symposium on Visual Analytics Science and Technology
October 24 - 28, Salt Lake City, Utah, USA
978-1-4244-9486-6/10/$26.00 ©2010 IEEE

allel Coordinates for class-based and non class-based datasets were presented.

Radviz is a radial visualization method, similar to Parallel Coordinates in the sense that it allows to visualize all dimensions of the dataset at once. It was first proposed in [9] to help the classification of DNA sequences. Later on, Radviz was extensively used to search for trends, especially clusters, in multidimensional datasets [10, 21, 17, 16]. We propose the use of quality measures to define an effective placement of the dimensions for a Radviz. Earlier, the dimensions were plotted either in the original order of the dataset or using a *Class Discrimination Layout Algorithm* [21]. This second method produces feasible results when applied to flattened datasets, i.e. the dimensionality is artificially expanded by splitting one dimension into two or more new dimensions [8], but our method is better suited for specific user tasks, like cluster searching (Section 3.3).

The second contribution of this paper is a quality measure to appraise the information content of projections in Pixel-Oriented Displays. Pixel-oriented visualization methods are very popular because they support the visualization of very large datasets. An overview of pixel-oriented visualization techniques is presented in [14]. A first trial on quality measures in such displays was proposed in [20], where an algorithm to measure the randomness of pixel visualizations was defined based on the entropy of the images. We propose a quality measure to appraise the information content of pixel visualizations. Our method was tested on Jigsaw maps [25] and was able to successfully rank the displays according to their overall information content.

Table Lens is a method to visualize large amounts of tabular data that uses a *focus & context* technique to display a detailed form of the data from selected table regions. It was first proposed in [19, 18], inspired by the *Generalized Fisheye Views* [7]. Later on, Bederson et al. applied the concepts from Table Lens in the *Date Lens fisheye calendar tool* [4]. The Data Lens was designed with the constrained display space of PDAs in mind and supports visualization of different time spans, search and presentation tools to highlight patterns and outliers. We propose an extension of the Table Lens method based on the information-bearing content on the table. Specifically, we use two measures to identify and highlight outliers and correlations between dimensions in a user defined Table Lens.

3 Radviz Analysis

Radviz[9] is a radial visualization method where the dimensions are represented by points placed equally spaced around a circumference. Each sample $\mathbf{x}_i$ of an n-dimensional dataset is represented by a point $\mathbf{p}_i$ in a 2-dimensional plot, as depicted in Figure 1. Imagine that each point $\mathbf{p}_i$ is connected by n springs to the n respective dimensions of the dataset and the spring constant K_i is equal to the j-th coordinate of $\mathbf{x}_i$, namely $x_{i,j}$. The final position of $\mathbf{p}_i$ in the visualization is determined by the point where the sum of all spring forces is zero and can be computed as:

$$\mathbf{p}_i = \frac{\sum_{j=1}^{n} \mathbf{d}_j x_{i,j}}{\sum_{j=1}^{n} x_{i,j}}, \qquad (1)$$

$\mathbf{d}_j$ is the vector pointing from the center to the position of the respective dimension on the circumference.

An important aspect of the Radviz visualization method is that it supports visualizing all dimensions of a dataset at once, such that it can be very useful while searching for clusters and outliers in high-dimensional data. Similar to Parallel Coordinates, a very important issue in Radviz is to decide in which order the dimensions shall be plotted to support a specific user task. Radviz is quite sensitive to the order of the dimensions, e.g. if dimensions with high values for a sample are placed close to each other in a sector on the circumference, this sample will be plotted towards this sector. Furthermore, samples with similar coordinate values are plotted close to the center.

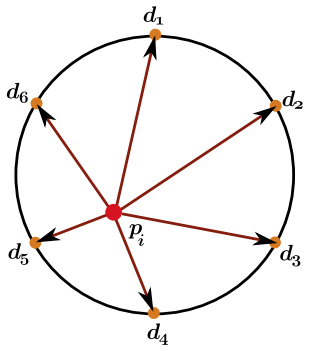


Figure 1: Radviz example. The dimensions j are represented by points, placed equally spaced around a circumference and each sample $\mathbf{x}_i$ is plotted at position $\mathbf{p}_i$ according to its coordinate values $x_{i,j}$

In this paper, we propose to make use of quality measures to generate a Radviz with an appropriate order of dimensions for a specific user task. A quality measure can be successfully applied to a visualization to appraise its information bearing content, but exhaustively computing all n-dimensional combinations in order to choose the best one requires a prohibitive amount of time for high-dimensional datasets.

Therefore the problem at hand is twofold. First, we need a useful quality measure to define whether a specific Radviz visualization provides useful information. This will be discussed in Section 3.1. Second, we need an efficient algorithm to guide the synthesis as it is unfeasible to create all possible visualizations. We will describe our approach in Section 3.2.

3.1 Quality Measures

Diverse quality measures can be used to quantify the amount of information of a Radviz, given a user task. Due to the scatter properties of a Radviz, most quality measures for Scatterplots may be applied to Radviz as well. In this paper, our analysis is based on the user task of searching for clusters, which is a very common task while visually exploring datasets with the Radviz method. For class-based datasets, we make use of the *Class Density Measure* (CDM) proposed in [23]. The CDM favors projections where the defined classes are well separated from each other and penalizes overlapping classes. For the non-class-based datasets, we propose a new quality measure to rank visualizations by searching for projections with well defined clusters. We call this measure *Cluster Density Measure*. Note that for our quality measures nomenclature, we assume clusters to be groups of data points close together in the visualization, while classes are defined as groups of data points with a previously know labeling.

3.1.1 Cluster Density Measure

The *Cluster Density Measure* (C_lDM) is designed to rank visualizations based on their clustering properties. Besides point-cloud-like visualizations, like Scatterplots and Radviz, it can also be directly applied to dense visualizations, like Continuous Scatterplots or Pixel-Oriented Displays. The C_lDM algorithm is directly applied to a visualization image and consists of two main parts: an image clustering algorithm; the measure estimation based on the cluster properties. Figure 2 gives an overview of the different steps of the algorithm.

For point-cloud-like visualizations (Figure 2(a)), we first compute a continuous representation of the image, whereupon a density

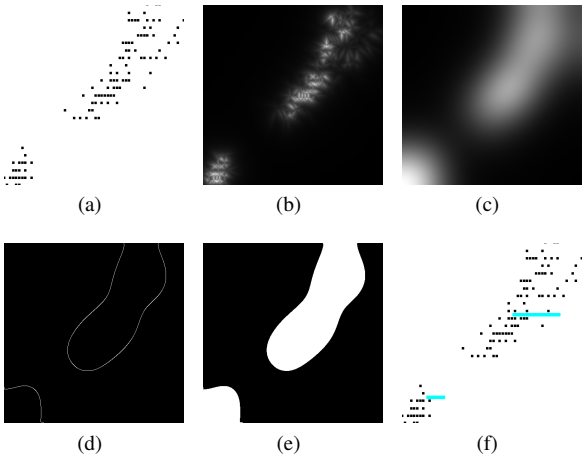
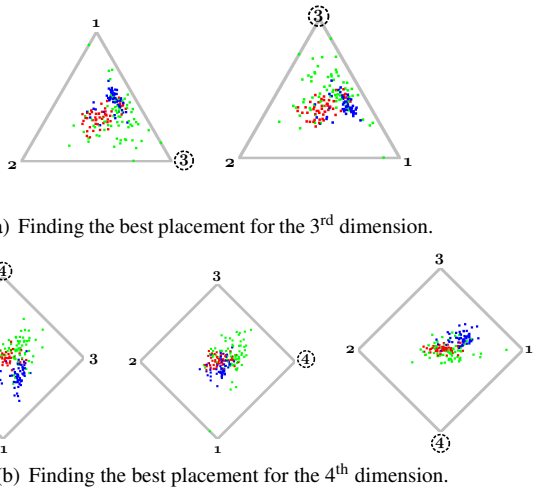


Figure 2: C$_l$DM algorithm example. (a) Original data plot, (b) density image computed based on local neighborhoods in the original image, (c) smoothed density image, (d) cluster contours obtained by zero crossing detection in the Laplace image of (c), (e) detected cluster regions and (f) original image with the average radius per cluster overlaid in the image.



(a) Finding the best placement for the 3rd dimension.

(b) Finding the best placement for the 4th dimension.

Figure 3: Dimension placement example for the first four dimensions of the *Wine* dataset. (a) The algorithm is initialized with the 1st and 2nd dimension and the best placement for the 3rd is computed. (b) Other dimensions are successively added and the best arrangement is kept in each step.

image (Figure 2(b)) is computed based on local neighborhoods in the original image. Working with a density image instead of the data points directly, has the advantage of not treating outliers as compact clusters. The density at a pixel $\mathbf{p}_i$ is defined as $1/r$, where r is the radius of the enclosing sphere of the k-nearest neighbors of $\mathbf{p}_i$ [23]. As these density images are usually still quite noisy, we extract the low frequency parts in order to create smooth density images (Figure 2(c)). This can be achieved by applying a Gaussian filter with a large standard deviation σ. The kernel width has a direct interpretation as the size of a region over which clusters should preferably not be mixed.

Clusters in these density images appear as smooth blobs (Figure 2(c)). We define the border of a cluster in the smooth density image as the point of inflection, i.e. where the curvature changes its sign. This can be conveniently found by convoluting the image with a Laplace filter and then searching for zero crossings (Figure 2(d)). Finally, we decide how many clusters are defined by the remaining contour based on the distance between the contour points. Two contour points are considered to belong to the same cluster if the distance between them is either smaller than a threshold τ or there exists a path along other contour points where the maximum distance between two adjacent points on the path is always smaller than τ. Otherwise, the contour points belong to different clusters. We set $\tau = \sqrt{P}/5$ where P is the number of pixels in the density image. After labeling the contours, the center $\mathbf{c}_k$ and average radius r_k per cluster are computed.The final measure is then defined as:

$$C_lDM = \frac{1}{K}\sum_{k=1}^{K}\sum_{l=k+1}^{K}\frac{d_{k,l}^2}{r_k r_l}, \qquad (2)$$

where K is the number of detected clusters and $d_{k,l}$ is the euclidian distance between the cluster centers $\mathbf{c}_k$ and $\mathbf{c}_l$. Accordingly, the C$_l$DM assigns high values to views that present well defined clusters with small intra-cluster distances and large inter-cluster distances.

3.2 Radviz Sorting

Given the CDM and C$_l$DM measure, we are able to quantify the quality of a Radviz plot. Now we need an efficient way to find a good Radviz plot without creating each possible visualization. We propose a greedy incremental algorithm to successively add dimensions to a Radviz plot to define a suitable order. This greedy approach provides a tradeoff between finding the optimal solution, which can be found by exhaustively searching all possible visualization arrangements, and completing all computations in a feasible time.

We start by creating a Radviz with only two dimensions. The first two dimensions added to the Radviz can be the first two in the dataset or the best two dimensions determined by a quality measure. We then add another dimension and create all possible 3D Radviz plots (at the current state only two positions are possible, see Figure 3(a)). According to the quality measure used, the best sequence of dimensions is selected for further processing. This intermediate Radviz is then successively augmented with all other dimensions by searching for and then keeping the best sequence with every dimension added. The final sequence defines a good placement of dimensions according to the chosen measure and user task.

Figure 3 shows an example of dimension placement for the first four dimensions of the *Wine* dataset. The algorithm is initialized with the 1st and 2nd dimension and the best placement for the 3rd is computed (Figure 3(a)). It is worth noting that a Radviz with three dimensions is not sensitive to their placement, the possible arrangements present only rotated and/or mirrored variations of the same structure. We then create all possible Radviz visualizations by adding the fourth dimension. The best ordering is kept. Therefore the overall complexity of the proposed algorithm is $O(n^2)$, which is comparably low to an $O(n!)$ exhaustive search.

3.3 Experiments

We tested our Radviz dimension placement algorithm on a variety of datasets. For classified datasets, we applied the *Class Density Measure* (CDM) [23] and for unclassified datasets the *Cluster Density Measure* (C$_l$DM) defined in Section 3.1.1. First, we show our results for the Wine dataset [1], a class-based dataset with 178 records and 13 dimensions that describes chemical properties of Italian wines from different cultivars. The first plot in Figure 4 is the original Radviz, without dimension replacement. In the second plot, the dimensions were reordered using the t-statistic algorithm proposed in [21]. In the third one the results of our placement algorithm are shown. The t-statistic is used to group dimen-

sions that similarly discriminate the records of a dataset and requires a classification of the records in some manner. Note that the t-statistic method presented a better plot than the original one, considering a cluster separation task, but the best plot for the same task was achieved using our method with quality measures. A second example, for another class-based dataset, is shown in Figure 5. *Olives*[29] is a classified dataset with 572 olive oil records from nine different regions from Italy, which define the classes of the dataset. For each sample, the normalized concentrations of eight fatty acids are given as attributes.

For unclassified data, we show our results in a synthetic dataset with ten dimensions, Figure 6. As the t-statistic method requires a classification of the data set, we compare our results for unclassified data only with the original Radviz. The left plot presented in Figure 6 is the original Radviz without any dimension reordering and the right one is the Radviz generated by our dimension placement algorithm. Observe that the resulting plot using our C$_l$DM method presents well separated clusters compared to the original plot.

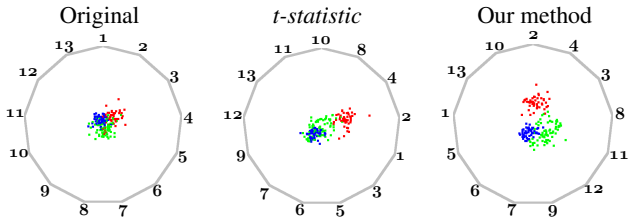


Figure 4: Original Radviz, t-statistic and our results, respectively, for the *Wine* dataset using the CDM measure. The different colors depict the different classes (cultivars) of the dataset.

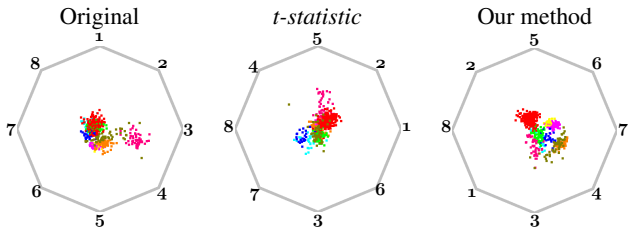


Figure 5: Original Radviz, t-statistic and our results, respectively, for the *Olives* dataset using the CDM measure. The different colors depict the different classes (regions) of the dataset.

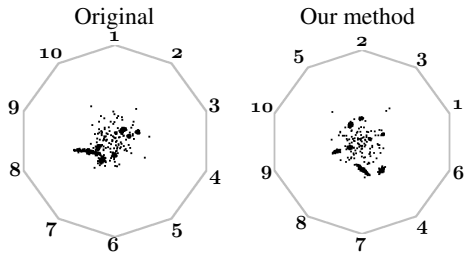


Figure 6: Original Radviz and our results, respectively, for a synthetic dataset with 10 dimensions and 8 clusters using the C$_l$DM measure.

4 Pixel-Based Displays

4.1 Jigsaw Maps

Standard projection techniques reduce the number of dimensions from n to two plus color information for visualization of the data on

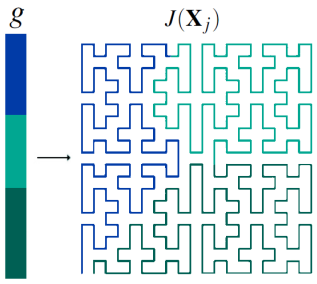


Figure 7: A screen-filling curve, here an H-curve, and a color mapping g is used to create a Jigsaw map.

the screen. In some cases, it turns out to be beneficial to go the other way around and represent a one dimensional function as a 2D plot in order to preserve some of the characteristics inherent in the data, e.g., natural order and locality. Examples of such data, would be the household income of a certain region or weather data. This is the idea behind Wattenberg's Jigsaw maps [25]. Jigsaw maps project the one dimensional data, or each dimension of multivariate data, into the 2D plane, using a space filling curve, in such a way that properties like locality and clusters are preserved, an example can be seen in Figure 7. If the dataset conveys more than one dimension, one Jigsaw map for each dimension is created.

An important step to create a Jigsaw map from a set of one dimensional data points $\mathbf{X}_j$ is to first normalize the values according to the desired output image size s^2: To serve this purpose a function $g(\mathbf{X}_j)$ needs to be defined to map $\mathbf{X}_j$ into sequences of subsets of $\{1,2,...,s^2\}$

$$g(\mathbf{X}_j) = (\{1,2,...,m_1\},\{m_1+1,...,m_2\},...,\{m_{k-1}+1,...,m_k\})$$

where $m_i = x_{1,j} + x_{2,j} + ... + x_{i,j}$, and $m_K = s^2$ where s^2 is the number of pixels in the output visualization. Width and height are considered to be of equal size and are usually a power of two. The layout function J for the Jigsaw maps can be defined using g and another function H: a screen-filling curve satisfying c-locality. C-locality is preserved wenn the diameter of a region r_i corresponding to a data point $x_{i,j}$ is bounded by a small constant c. This keeps regions relatively compact in the output. Then

$$J(\mathbf{X}_j) = H(g(\mathbf{X}_j)) \ ,$$

i.e. each data value is given a set of connected positions along the screen filling curve and a color. A common choice for the color mapping is to assign the data values to a color gradient in order to ease the interpretation of the underlying data. We use linear mapping for the experiments in this paper. However, other sophisticated mappings [5] may be used without additional effort together with our quality measures. An example for a space-filling curve and its colorization is given in Figure 7.

4.1.1 Quality Measure

The task at hand is to find interesting structures in a Jigsaw map, clusters are the most well known. But it can be hard to find clusters, as these structures usually do not have any specific size or layout which makes it difficult to describe them in a mathematical sense. Schneidewind *et al.* [20] used the entropy or standard deviation of the color values in different grid cells to derive a quality measure. The algorithm considered regions as interesting if the entropy or standard deviation was larger than a certain threshold τ_{min} but smaller than another threshold τ_{max}. There are two drawbacks of this method. First, the thresholds need to be set by hand.

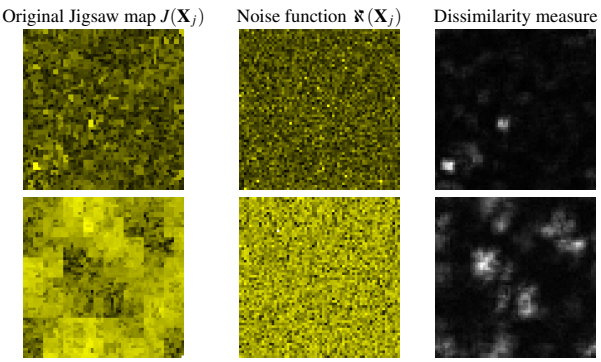


Figure 8: Visualization of the noise dissimilarity, from left to right: Original Jigsaw map $J(\mathbf{X}_j)$, corresponding noise function $\aleph(\mathbf{X}_j)$, our noise dissimilarity measure computed for each pixel (values are linearily scaled for better readability). Lighter values correspond to a stronger dissimilarity and therefore more interesting structures. The top row shows an outlier example in the *Ozone* dataset [28]. Our algorithm highlights these outliers as interesting regions. In the bottom row a visualization with more abstract patterns is shown. Note that most of these structures are captured well with our approach.

Without any knowledge of the underlying data this can be tedious. Second, entropy as well as standard deviation do not pay any attention to the spatial arrangement of the data. To do so, the user is urged to provide a weighting function for the hierarchical analysis in [20]. Again this can be difficult, if the structures searched for are unknown.

Instead of searching for interesting structures directly, we propose to quantify the *dissimilarity* to a noise function $\aleph(\mathbf{X}_j)$. We impose $\aleph(\mathbf{X}_j)$ to have the same color probabilities as $J(\mathbf{X}_j)$, i.e. the histograms of both images are equal. In practice we can easily achieve this by randomly permutating the pixel coordinates of $J(\mathbf{X}_j)$. An example of such a Jigsaw map $J(\mathbf{X}_j)$ and its corresponding noise function $\aleph(\mathbf{X}_j)$ is given in Figure 8 in the left and middle images.

Next, we assume that Jigsaw maps $J(\mathbf{X}_j)$ as well as the noise function $\aleph(\mathbf{X}_j)$ can be modeled by a Markov Random Field that models the images as a realization of a local and stationary random process. Each pixel is characterized by a small set of spatially neighboring pixels, and this characterization is the same for all pixels. This model has been succesfully used for exemplar-based texture synthesis, see e.g. [26], which is related to our problem. In exemplar-based texture synthesis one starts with a noise function and tries to explain this noise function with a given input image. Now, we do the same but with exchanged images. We will try to explain a given visualization $J(\mathbf{X}_j)$ with potential information by a given noise function $\aleph(\mathbf{X}_j)$.

Due to the assumed locality and stationary characteristics of our images, we can base our quality measure on the similarity, respectively the dissimilarity, between local neighborhoods of each pixel in $J(\mathbf{X}_j)$ and each pixel in $\aleph(\mathbf{X}_j)$. We call this the *Noise Dissimilarity Measure* (NDM). We denote a neighborhood of pixels with radius r around a pixel i with $J_{NH(i)}(\mathbf{X}_j)$, $\aleph_{NH(i)}(\mathbf{X}_j)$. To quantify the noise dissimilarity we compute:

$$NDM(J(\mathbf{X}_j),\aleph(\mathbf{X}_j)) = \frac{1}{\omega}\sum_{i=1}^{s^2} diss(J_i(\mathbf{X}_j),\aleph(\mathbf{X}_j)) \,, \qquad (3)$$

with

$$diss(J_i(\mathbf{X}_j),\aleph(\mathbf{X}_j)) = \min_k(||J_{NH(i)}(\mathbf{X}_j)-\aleph_{NH(k)}(\mathbf{X}_j)||^2)$$

$$\text{and } \omega = s^2(2r+1)^2$$

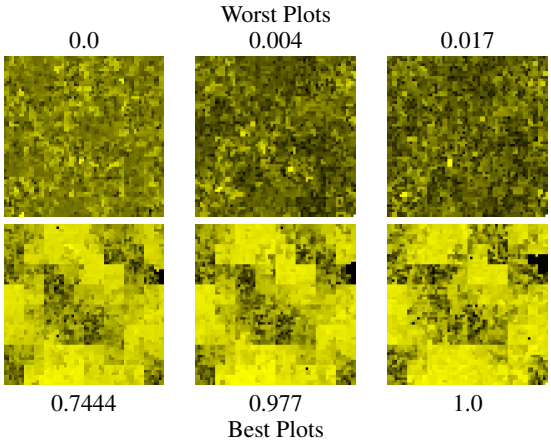


Figure 9: Jigsaw maps[25] of the *Ozone* dataset [28]: The top row shows the three worst plots, while the bottom row shows the three best plots and their associated normalized goodness values as it was estimated by our NDM. Obviously the top row contains mainly noise and it is difficult to find interesting regions in such a visualization, also note that its overall appearance is more dull. The best plots according to the NDM show a larger degree of potentially meaningful patterns and a higher contrast. The goodness values of all plots have been normalized to range between 0 (for the worst plot) and 1 (for the best plot).

Here $diss(J_i(\mathbf{X}_j),\aleph(\mathbf{X}_j))$ is simply the sum-of-squared differences between a neighborhood vector NH around pixel position i in $J(\mathbf{X}_j)$ and the best matching neighborhood in $\aleph(\mathbf{X}_j)$ which was found at pixel k. The size of the neighborhood is defined by its radius r. ω is a normalization factor which makes the measure invariant towards the image size and the neighborhood radius.

As the neighborhood matching employed in calculating the NDM is a very costly procedure, we apply different techniques to speed up the process. We limit the radius of the neighborhoods to $r=2$, resulting in a 5×5 neighborhood, which turns out to be sufficient in our test cases. As we use color images, this results in a 75D vector for comparison. We accelerate neighborhood matching by projecting the 5×5 pixel neighborhoods into a truncated 12D principal component analysis (PCA) space. In addition we use a fast nearest-neighbor search [2] to find the best matching neighborhood in $\aleph(\mathbf{X}_j)$.

The NDM has several beneficial properties. The characteristic of the noise function is a total absence of structures, therefore regions in $J(\mathbf{X}_j)$ differing from every neighborhood in $\aleph(\mathbf{X}_j)$ will likely contain interesting patterns. In addition, this measure also penalizes badly chosen color mappings. Low contrast images will result in less dissimilarity to $\aleph(\mathbf{X}_j)$, while high contrast images are preferred.Theoretically, the NDM should be applicable with little changes to other visualization methods, e.g. Pixel Bar Charts [15], but we currently did not investigate further in this direction.

4.1.2 Experiments

As an application example we analyzed the *Ozone Level Detection* dataset [28] with 2536 instances and 73 dimensions. The visualizations for dimension 9 (WSR8), 24 (WSR23) and 60 (U70) have been considered to be the worst by our algorithm. They hardly contain any interesting regions and mainly contain noise (Figure 9, top row). Dimension 9 and 24 depict the measured wind speeds at different times, which are relatively constant with only few changes. On the other hand, dimensions 30 (T3), 34 (T7) and 35 (T8), which depict the temperature at 3am, 7am and 8am in the morning, provide insights into the change of temperature throughout the years.

5 TABLE LENS

Graphical Compressed Tables (GCT) are a common technique to visualize large datasets in tabular form. Within this representation columns contain the information for the different dimensions and rows correspond to the attributes or samples of the dataset. Every row and attribute has a height of one pixel and represents the underlying data qualitatively utilizing a tiny bar metaphor. Similar to other visualization techniques, only the relative values are visible for each dimension. Row-based sorting techniques allow for recognizing dimensional-dependencies (for instance correlation) in a comprehensive manner.

Table Lenses (TL) are an established and well-accepted *focus & context* technique for GCT's and have been introduced in [19] and [18]. Using this technique, the user can select dimensions (column) and attributes (rows), that are then represented as spreadsheets, instead of the compressed style. Thus, it is possible to steer the Degree of Interest (DOI) and allow for systematic explorative visual search. A subset of information is selected from the dataset by selecting dimensions and attributes in a GCT representation.

Table Lenses easily become confusing for the user as soon as more than a few dimensions or samples have been selected and it becomes difficult to see interesting patterns like correlation or outliers in the data. An example of a GCT and an according GCT+TL is shown in Figure 10.

In this section, we present a semi-automatic approach to extend the Table Lens technique to visually support and present inter- and intra-dimensional relations in the selected data. We support user tasks like outlier detection or correlation detection. The user can select regions which are enhanced similar to a Table Lens, but additional information from the applied quality measures are added to support the analysis.

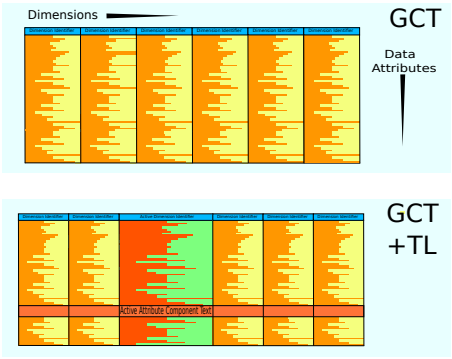


Figure 10: Standard Design for a Graphical Compressed Table and the Table Lens extension. Top row: Dimensions are mapped to columns, with a dimension descriptor at the top, while the different data values are mapped to the corresponding row in a GCT. Bottom row: GCT with active TL emphasize a selected subset of the underlying dataset.

5.1 Quality Measures

5.1.1 Data extraction

In the special case where the original data is no longer available, the GCT must be analyzed directly to extract the relative data values for each dimension and sample. We will not go into too much detail here, but basically one can search for horizontal and vertical lines in the same color as the border and extract them. This divides the image into different grid cells. After removing the dimension identifier at the top of each column, each grid cell left represents one dimension of the dataset. As each attribute is one line in these cells, the relative (and potentially discretized) data values can be easily extracted.

5.1.2 Attribute Correspondence Matrix

Once we reconstructed the dataset up to a scale factor for each dimension, the user can start his/her visual analysis by selecting certain dimensions and attributes, similar to Table Lenses. From the selected regions we extract an *attribute correspondence matrix D*. Each column of D represents one of the selected dimension d_{id}, while each row represents the selected attributes a_{id}, see Figure 11.

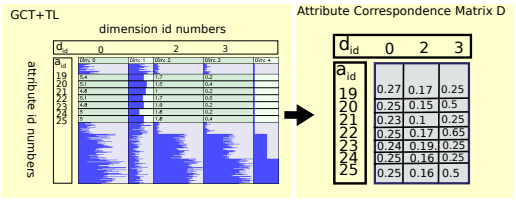


Figure 11: Example of the attribute correspondence matrix D. The matrix is established from the selected dimensions, here 0,2 and 3, and the selected attributes, 19 till 25, and the corresponding values from the GCT (for better readability the GCT+TL is shown).

5.1.3 Enhanced Table Lens Visualization

Using the attribute correspondence matrix D, a local statistic data analysis can be performed on the basis of the extracted qualitative values. To demonstrate this procedure, we describe algorithms applied to D to identify outliers and linear correlations. In order to find a suitable measure to describe outliers, all attribute values are assumed to follow a normal distribution. For each value $D(d_i,a_j)$, a value $o(d_i,a_j)$ is calculated that describes the likeliness for an outlier:

$$o(d_i,a_j) = min(1, \frac{|D(d_i,a_j) - \mu_{d_i}|}{3\sigma})$$

with

$$\mu_{d_i} = \frac{1}{n_{aa}} \sum_{l=0}^{n_{aa}-1} D(d_i,a_l); \quad \sigma = \sqrt{\frac{1}{n_{aa}} \sum_{l=0}^{n_{aa}-1} (D(d_i,a_l) - \mu_{d_i})^2},$$

where n_{aa} is the number of active attributes. If $o(d_i,a_j) > 3\sigma$ then $o(d_i,a_j)$ is 1. This choice is motivated by the fact that 99.97% of all values are within a $\mu \pm 3\sigma$ environment (3 sigma rule). The closer a value $D(d_i,a_j)$ is to the boundary of 3σ, the more likely this value can be considered as an outlier. To highlight these outliers visually, we use a colormap to encode the corresponding area in the GTC+TL relative to $o(d_i,a_j)$. High values are marked yellow and low ones are colored in dark green, see Fig. 12 middle right. For each possible pairwise combination of dimensions d_i and d_j in D the *correlation coefficient* $k(d_i,d_j)$ is calculated to emphasize the linear correlation between two dimensions of the subset:

$$k(d_i,d_j) = \left| \frac{\sum_{l=0}^{n_{aa}-1} (D(d_i,a_l) - \mu_{d_i})(D(d_j,a_l) - \mu_{d_j})}{\sqrt{\sum_{l=0}^{n_{aa}-1} (D(d_i,a_l) - \mu_{d_i})^2 \sum_{l=0}^{n_{aa}-1} (D(d_j,a_l) - \mu_{d_j})^2}} \right|$$

with

$$\mu_{d_i} = \frac{1}{n_{aa}} \sum_{l=0}^{n_{aa}-1} D(d_i,a_l)$$

Due to the absolute value of k, our measure also detects negative correlations.

The n best correlation pairs, i.e. pairwise dimensions, will be visualized in the Table Lenses. This is done by mapping every single combination of the best pairs (d_i and d_j) to an individual color. The color of one pair implicitly encodes a visual relation between these dimensions. While the base color can be chosen arbitrarily, the brightness b of that color is defined as:

$$b = \frac{k(d_i, d_j)}{k_{\max}} \tag{4}$$

with

$$k_{max} = max_{i,j}(k(d_i, d_j))$$

Using this description, the individual correlation coefficient modifies the brightness of the final color, highlighting (relatively) good correlations. The corresponding TL area is split into s equally-colored sectors, where s is the number of correlations between dimension d_i to the other marked dimension which are among the n best correlation coefficients, see Fig. 12 on the right.

5.2 Experiments

Figure 12 depicts the experimental results for the *Olive* dataset [29] and the *Yeast* dataset [11] with different Table Lens configurations. In the leftmost image the GCT is shown followed by the standard GCT with TL. The image on the middle right illustrates our result on potential outlier detection (yellow) and the image on the right marks pairwise linear correlations for the three best correlation candidates within the subset.

It can be stated that outliers can be identified fast and are well recognizable using the imposed color scheme. Less probable candidates are shown in more decent colors. In addition to this, the correlation between different active parts of the dimensions can be detected quickly. For instance the visualization reveals that dimension 2 and 5, 2 and 6 as well as 5 and 6 are in direct relation to each other in the *Olive* dataset shown in Figure 12 (a).

The significance of the identified linear correlation can be visually analyzed using the displayed color brightness. It can be observed that the selected regions of dimension 2 and 5 (light green) has the strongest correlation within this configuration. Furthermore, it is noticeable that dimension 2 and 3 have the strongest correlation within the selected subset of the *Yeast* dataset (Fig. 12 (b)).

Note that the correlations are still relative and not absolute with respect to the original dataset. Therefore the visualization enhances correlation tendencies within the given subset without communicating the actual absolute strength of those dependencies. In general an absolute quantization of the analyzed values can be deduced by setting $k_{\max} = 1$ in equation 4, which removes the value normalization for b.

6 Conclusion

In this paper, we extended the existing set of quality measures presented in [23], and proposed new techniques for three other popular visualization methods: Radviz, Pixel-Oriented Displays and Table Lens. Specifically, we presented an improvement for the Radviz method with a greedy dimension placement algorithm based on quality measures. This can be applied to data with predefined categorical labels or datasets without any class information, We compare our method with a previously proposed sorting method for Radviz [21] and show the improvements introduced by our approach. Moreover, we proposed a new quality measure specialized for the clustering user task in point cloud-like visualization methods, e.g. Scatterplots and Radviz, an algorithm to detect information-bearing structures in Pixel-Oriented Displays, and finally, we incorporated information-bearing algorithms into the Table Lens technique for the purpose of finding correlation and outliers. With respect to performance, in our examples, the computation time has never exceed 300 milliseconds for the Table Lens algorithm, 180 milliseconds for each Jigsaw image, and 100 seconds for the Radviz sorting using the CDM measure. However, this computing time may be significantly reduced using a GPU implementation.

The presented contributions are able to aid and potentially speed up the visual exploration process and are a further step towards the realization of an effective and efficient visual analysis tool for high-dimensional data. As future work, we intent to compare quality measures for different visualization methods and connect all these approaches in a single visualization tool.

Acknowledgements

This work was supported in part by a grant from the German Science Foundation (DFG) from projects DFG MA2555/6-1 and DFG TH692/6-1, within the strategic research initiative on Scalable Visual Analytics.

References

[1] S. Aeberhard, D. Coomans, and O. D. Vel. Comparative-analysis of statistical pattern-recognition methods in high-dimensional settings. pattern recognition. In *IEEE Signal Processing Workshop on Higher Order Statistics.*, pages 14–16. John Wiley, 1994.

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 573–582, 1994.

[3] D. Asimov. The grand tour: a tool for viewing multidimensional data. *Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.

[4] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson. Datelens: A fisheye calendar interface for pdas. *ACM Transactions on Computer-Human Interaction*, 11:90–119, 2004.

[5] E. Bertini, A. D. Girolamo, and G. Santucci. See what you know: Analyzing data distribution to improve density map visualization. In *EuroVis*, pages 163–170, 2007.

[6] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

[7] G. W. Furnas. Generalized fisheye views. *SIGCHI Bull.*, 17(4):16–23, 1986.

[8] G. Grinstein, P. Hoffman, S. Laskowski, and R. Pickett. Benchmark development for the evaluation of visualization for data mining. In *Information Visualization in Data Mining and Knowledge Discovery*, pages 129–176. Morgan Kaufmann, 2001.

[9] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *Proceedings of the 8th conference on Visualization '97*, pages 437–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.

[10] P. Hoffman, G. Grinstein, and D. Pinkney. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In *NPIVM '99*, pages 9–16, New York, NY, USA, 1999. ACM.

[11] P. Horton and K. Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. In *International Conference on Intelligent Systems for Molecular Biology*, pages 109–115, 1996.

[12] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[13] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, December 1985.

[14] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6:59–78, 2000.

[15] D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: A visualization technique for very large multi-attribute data sets. *Visualization, extended version in: Information Visualization Journal, Palgrave*, 1(2), 2002.

[16] L. Nováková and O. Štěpánková. Radviz and identification of clusters in multidimensional data. In *IV '09: Proceedings of the 2009 13th International Conference Information Visualisation*, pages 104–109, Washington, DC, USA, 2009. IEEE Computer Society.

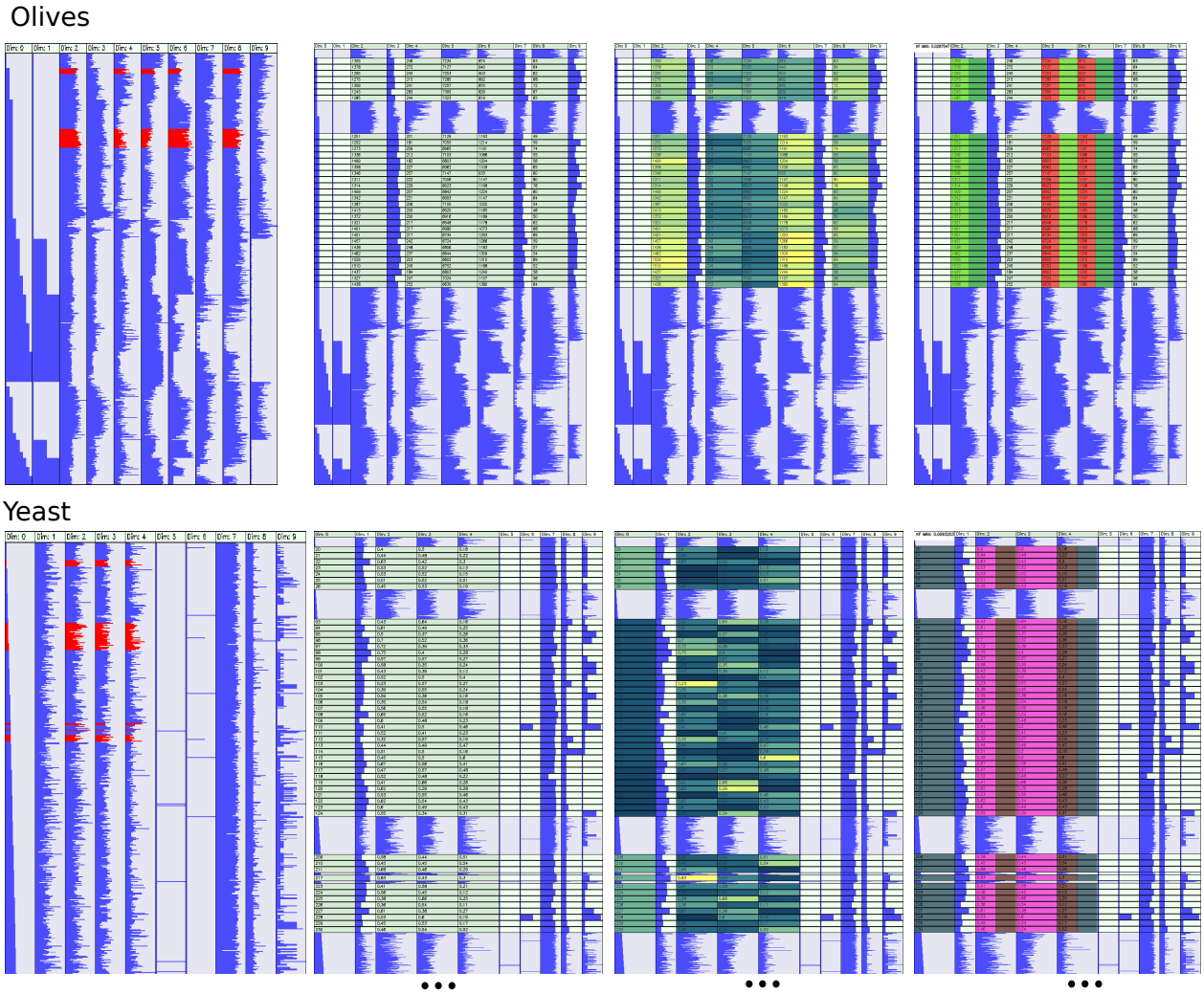


Figure 12: Overview of the visually enhanced Table Lens results for the *Olive* dataset (top) and *Yeast* dataset (bottom). From left to right: Original Graphical Compressed Table (we marked the region the user wants to enhance in red), GCT plus Table Lens, our visual enhancement for outlier detection and our visual enhancement for correlation estimation.

[17] L. Nováková and O. Štěpánková. Visualization of trends using radviz. In *ISMIS '09: Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, pages 56–65, Berlin, Heidelberg, 2009. Springer-Verlag.

[18] P. Pirolli and R. Rao. Table lens as a tool for making sense of data. In *AVI '96: Proceedings of the workshop on Advanced visual interfaces*, pages 67–80, New York, NY, USA, 1996. ACM.

[19] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. pages 318–322, Boston, Massachusetts, USA, 1994. ACM.

[20] J. Schneidewind, M. Sips, and D. Keim. Pixnostics: Towards measuring the value of visualization. *Symposium On Visual Analytics Science And Technology*, 0:199–206, 2006.

[21] J. Sharko, G. Grinstein, and K. A. Marx. Vectorized radviz and its application to multiple cluster datasets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1444–1427, 2008.

[22] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum (Proc. EuroVis 2009)*, 28(3):831–838, 2009.

[23] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST)*, Atlantic City, New Jersey, USA, 10 2009.

[24] J. Tukey and P. Tukey. Computing graphics and exploratory data analysis: An introduction. In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics 85. Nat. Computer Graphics Assoc.*, 1985.

[25] M. Wattenberg. A note on space-filling visualizations and space-filling curves. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 24, Washington, DC, USA, 2005. IEEE Computer Society.

[26] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009.

[27] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 157–164, 2005.

[28] K. Zhang and W. Fan. Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond. *Knowl. Inf. Syst.*, 14(3):299–326, 2008.

[29] J. Zupan, M. Novic, X. Li, and J. Gasteiger. Classification of multicomponent analytical data of olive oils using different neural networks. In *Analytica Chimica Acta*, volume 292, pages 219–234, 1994.