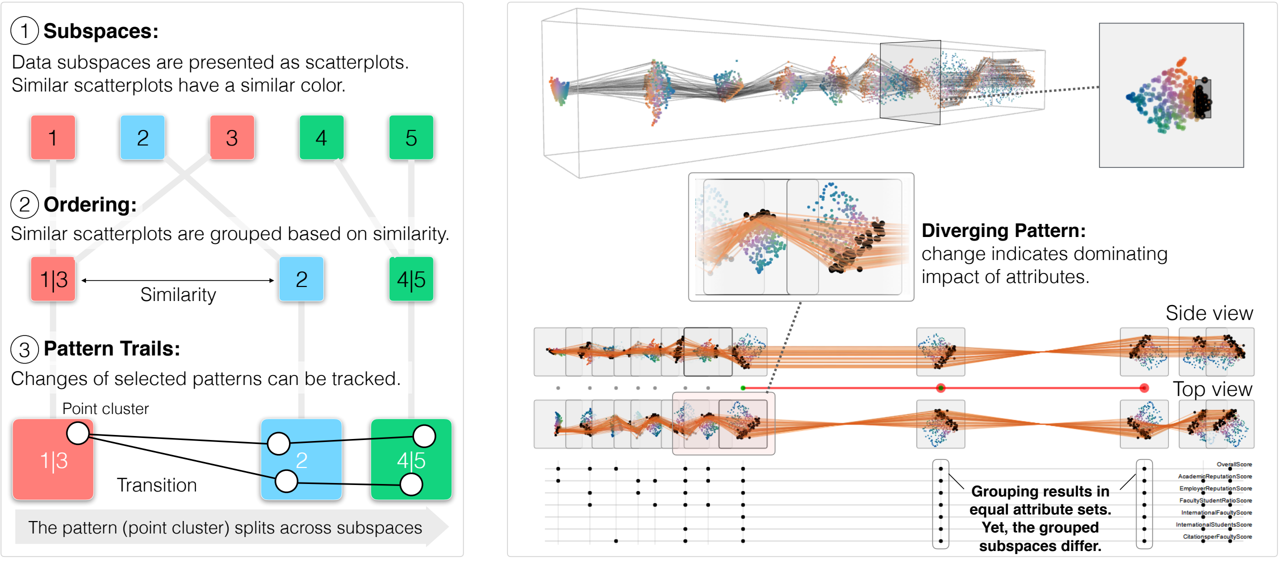
Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces

Dominik Jäckle* University of Konstanz
Michael Hund† University of Konstanz
Michael Behrisch‡ University of Konstanz
Daniel A. Keim§ University of Konstanz
Tobias Schreck¶ TU Graz


1 Subspaces:
Data subspaces are presented as scatterplots. Similar scatterplots have a similar color.
1
2
3
4
5
2 Ordering:
Similar scatterplots are grouped based on similarity.
1|3
Similarity
2
4|5
3 Pattern Trails:
Changes of selected patterns can be tracked.
Point cluster
1|3
Transition
2
4|5
The pattern (point cluster) splits across subspaces
Diverging Pattern:
change indicates dominating impact of attributes.
Side view
Top view
Grouping results in equal attribute sets. Yet, the grouped subspaces differ.
OverallScore
AcademicReputationScore
EmployerReputationScore
FacultyStudentRatioScore
InternationalFacultyScore
InternationalStudentsScore
CitationsperFacultyScore

(a) (b)

Figure 1: Visual analysis of subspace patterns by a series of consecutive pattern transitions between scatterplots. (a) Scatterplots depict subspaces and are grouped and sorted based on similarity. (b) This example shows the pattern transitions in the University data set. Based on a 3D cube like visualization, one can trace sorted patterns in a side and top view on the cube (see Section 7.1).

Abstract

Subspace analysis methods have gained interest for identifying patterns in subspaces of high-dimensional data. Existing techniques allow to visualize and compare patterns in subspaces. However, many subspace analysis methods produce an abundant amount of patterns, which often remain redundant and are difficult to relate. Creating effective layouts for comparison of subspace patterns remains challenging. We introduce *Pattern Trails*, a novel approach for visually ordering and comparing subspace patterns. Central to our approach is the notion of *pattern transitions* as an interpretable structure imposed to order and compare patterns between subspaces. The basic idea is to visualize projections of subspaces side-by-side, and indicate changes between adjacent patterns in the subspaces by a linked representation, hence introducing pattern transitions. Our contributions comprise a systematization for how pairs of subspace patterns can be compared, and how changes can be interpreted in terms of pattern transitions. We also contribute a technique for visual subspace analysis based on a data-driven similarity measure between subspace representations. This measure is useful to order the patterns, and interactively group subspaces to reduce redundancy. We demonstrate the usefulness of our approach by application to several use cases, indicating that data can be meaningfully ordered and interpreted in terms of pattern transitions.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces;Interaction Styles

1 Introduction

Information is collected at large-scale in all areas of our day-to-day life: statistical surveys of natural disasters or inhabitants, rankings of public institutions, or any tabular data that consists of multiple observations and attributes. A main task in understanding such multivariate data is to identify and interpret relevant patterns like dense groups (clusters), outliers, or correlations. Often, data comprises many attributes (high-dimensional data), and relevant patterns are not only found in the full attribute space, but also among subspaces – we refer to subspaces as attribute subsets of the data. More importantly, especially in high-dimensional data, patterns may only be found in smaller subspaces and would get lost when considering all dimensions at once [6]. However, we cannot assume that patterns will be similar across different subspaces. Rather, we can expect they may be structurally different in different subspaces, posing the challenge to identify, interpret, and compare them visually. Visual analytics suggests to involve the analyst into the automated analysis process using interactive data visualizations [43]. By leveraging the human capabilities to explore the data, visual analytics facilitates finding relevant patterns and fosters sensemaking.

Automatic subspace analysis and clustering methods provide sets of possibly interesting patterns and subspaces. While such methods drastically reduce the amount of possible attribute configurations by ignoring subspaces with a high attribute and pattern overlap, they entirely leave out the analyst [31] from the initial search process. They typically provide no hints on an appropriate ordering or on relationships among the reported subspaces. Hence, the amount of considered subspaces can in fact be reduced, yet it is challenging to explore the data to find interesting patterns. In recent years, several

*e-mail: Dominik.Jaeckle@uni-konstanz.de
†e-mail: Michael.Hund@uni-konstanz.de
‡e-mail: Michael.Behrisch@uni-konstanz.de
§e-mail: Keim@uni-konstanz.de
¶e-mail: Tobias.Schreck@cgv.tugraz.at

2017 IEEE Conference on Visual Analytics Science and Technology (VAST)
October 1–6, Phoenix, Arizona, USA
978-1-5386-3163-8/17/$31.00 ©2017 IEEE

approaches have been presented to visually explore multivariate data, and in particular patterns in subspaces. Parallel Coordinate Plots (PCPs) [22] present a key technique for multivariate data analysis. Besides researched challenges such as axis re-ordering, one axis merely corresponds to one attribute making it a difficult task to compare different attribute combinations with each other. Recent work proposes to augment PCPs with Scatterplots [56]. Scatterplots are a prominent means to make subspaces visually accessible and can enhance the analysis of subspace patterns. However, this approach proposes to manually switch between the representations, making it a tedious, challenging task to search for interesting subspaces and patterns. Scatterplots can also be combined to Scatterplot Matrices (SPLOMs) that enable pairwise comparisons between attributes, effective for small to moderate sized data sets [46].

Recent advances in machine learning propose Dimensionality Reduction (DR) to transform the data to a lower-dimensional space, preserving the main structure of the data. We refer to the structure of the data as interrelated or correlated subsets of attributes over subsets of data records. Results can be depicted in a two-dimensional scatterplot, in which proximity between points indicates similarity. Techniques using DR typically present only one view on the data or present all subspaces via small multiples [52], making it challenging to identify interesting patterns and trace their meaning in different subspaces. Even if applying subspace analysis prior to the visual exploration phase, subspaces can still be redundant and too many to identify relevant patterns. Automatic approaches require the user to adapt the analysis model based on the task at hand to retrieve relevant views on the data. *Pattern Trails* aims to address the unsettled question: *How to identify and relate interesting patterns among multivariate subspaces, using interactive visual exploration?*

Pattern Trails is an interactive visual approach for the exploration of subspaces of multivariate data. Its main goal is to find and explain pattern transitions among subspaces. Pattern transitions can, for example, indicate structural changes in a pattern caused by dominating attributes. We first apply automatic subspace analysis to reduce the sheer amount of possible subspaces to interesting ones. Then, we apply DR, in particular, distance-preserving projections, to make each subspace visually accessible. The projected subspaces are depicted in a small-multiple [52] environment using scatterplots. Furthermore, we highlight transitions between the subspace depictions for tracing structural changes of patterns. A transition connects a pattern among two subspaces by individual lines each line corresponding to one data record. To enable understanding of patterns and pattern transitions, we make two contributions: First, we provide a *systematization and categorization of pattern transitions among subspaces of multivariate data*. We introduce a pattern *trail* as a set of pairwise transitions between subspaces. A trail visually connects data records that form a pattern across all subspaces along which the patterns can be meaningfully compared. Thereby, various transition types occur, each having a different meaning. Second, we provide a *data-driven similarity measure for projections to group subspaces and overcome redundancy*. Because data sizes and tasks differ, we tightly couple this process and the user who steers the parameters to obtain an effective aggregation of scatterplots, and thus subspaces.

We integrate the systematization of pattern transitions and the user-steered similarity between subspaces into our visual approach. Our visual approach consists of a horizontal view on the objects and a vertical view on the contributing attributes. The view on both spaces supports the interpretation and understanding of patterns.

2 Related Work

Patterns trails supports the visual analysis of subspace patterns by ordering and relating patterns, hence support users to explain structural changes of patterns among different subspaces. Subspace analysis and visualization are subtopics of high-dimensional data analysis that have recently gained research interest. The term *high-dimensional* is understood as data with many dimensions (synonym for attributes), yet we apply our methods to data with up to 8 dimensions. As we do not consider this amount as high-dimensional, we keep to the general notion of *multivariate data*, but comment that the subspace set is large even for 8 dimensions as 2^8 (powerset), resulting in an expensive, exhaustive search. We discuss related work from multivariate data analysis, DR for visual analysis, and subspace search and visualization.

2.1 Multivariate Data Analysis and Visualization

Multivariate data typically consists of several interdependent attributes [39]. The main goal is to find discernible patterns to draw conclusions about the structure of and to gain insight into the data. We thereby distinguish between automatic and visual approaches. Automatic methods, such as methods from data mining or machine learning, search for patterns in the form of clusters or classes [18,41]. Resulting patterns are then either presented quantitatively or are explicitly enumerated as sets of data entries. However, it is challenging to make sense of results that do not fit the task at hand or are large-scale. This is where interactive visualization acts as a means to make the data accessible to the user and foster sensemaking [14].

In recent years, several approaches have been presented to visualize multivariate data. Common approaches include geometric projections (Andrew Curves [3], Parallel Coordinate Plots [22]), pixel-based techniques (Recursive Patterns [28], Pixel Bar Charts [29]), glyph-based techniques (Star Glyphs [9], or Chernoff Faces [11]). For further reading, we refer to the comprehensive surveys carried out by Kehrer and Hauser [27] or Liu et al. [36]. Named techniques are prominent for making an attempt to visualize all data in one display encoding one up to all attributes. However, patterns are often only given in subspaces of the data, which are specific attribute sets, and are not discernible in a global representation [6]. In the following, we discuss the visualization and analysis of multivariate subspaces to find relevant patterns.

2.2 Using Dimensionality Reduction for Visual Analysis

Multivariate data consists of multiple attributes, posing a challenge to identify expressive ones that reveal interesting patterns. DR, therefore, transforms the data to a lower-dimensional space. Results are typically presented in a two-dimensional scatterplot where proximity between points indicates how similar they are. Known DR methods that enable visual analysis include, but are not limited to, linear methods such as Multidimensional Scaling (MDS) [12] and Principal Component Analysis (PCA) [26] or non-linear methods like t-distributed stochastic neighbor embedding (t-SNE) [38] and Self-Organizing Maps (SOM) [30].

According to Sacha et al., interaction enables exploratory data analysis of DR results and adapts to the “human needs and domain-specific problems” [44, p. 214]. First interactive approaches to DR include the pioneering work by Buja and Cook [2] and Ward and Martin [54]. Recent works integrate interaction to steer the projection and analyze the attribute space to make sense of salient structures. Recent techniques include iPCA [24], Dimstiller [21], Brushing Dimensions [53], Data Context Map [10], and Probing Projections [47]. Additional interactive approaches can be found in the survey of Sacha et al. [44]. Interactive techniques for steering and analyzing projections of multivariate data are efficient to analyze the data at a global scale and to interpret selected patterns. However, patterns often only occur in subspaces of the data. Using global views, it remains challenging to identify relevant subspaces.

2.3 Subspace Search and Visualization

Scatterplots are practically the first choice to depict the results of DR. Proximity between points indicates how similar they are. Computing similarity or proximity between data, or reducing their dimensionality, is more difficult to do as more attributes are introduced. Typically, the more attributes are introduced, the less discriminative the projection result is. As a result, it is challenging to identify patterns, also known as the *curse of dimensionality* [6]. Kriegel et al. reason that as the number of attributes grows, the relevance of attributes can differ for different patterns [33]. In other words: Not all attributes contribute to the existence of a pattern, but relevant patterns exist in different combinations of attributes, namely the subspaces.

Recent advances in visualization build on top of automatic subspace analysis [33] or clustering approaches [42], and make the result accessible for exploration. In doing so, the visualization of subspaces is either applied to the attribute space of the data, the object space, or to both combined. The attribute space refers to the general statistical analysis of the attributes that comprise a given subspace. For instance, the approaches of Krause et al. either provide a sorted frequency-based view on the attribute values [31], or enable finding relevant attributes (features) based on feature rankings [32]. In contrast, object views present the entities of the data set and allow, typically in combination with analysis possibilities including the attributes, to foster understanding of single subspaces or patterns. Liu et al. [37], for example, provide a comprehensive interactive view on the projection. Other approaches combine the object and the attribute view on the data [20, 50, 51, 57, 58].

Purely object-driven approaches are related to the field of projection pursuit [19], where the overall goal is to find significant projections of multivariate data; these are projections where points build unique patterns. Examples include the work by Anand et al. [1] and Lehmann and Theisel [34]. Related works on multivariate cluster visualization [8] and comparison [7] do not build on subspace analysis. They focus on the presentation of relevant features to build clusters and enable comparison. There are further related approaches, but in summary, they do not scale for many subspaces.

The commonality between aforementioned approaches is, that they either are overstrained by the number of subspaces, miss support for pattern ordering and comparison or produce an abundant amount of patterns, which often remain redundant and difficult to relate. Existing systems typically impose small-multiple [52] alike views on the data with some ordering of the views, but do not provide an ordering and linking dedicated to comparing subspaces with regard to which data points change and which remain stable across the subspaces. Also, it is not enough to show interesting patterns across subspaces. Identified patterns require practical support to *explain* their occurrence concerning their structural change and the attribute configuration. In this paper, we improve visual subspace analysis by imposing meaningful ordering on subspaces, used to group similar subspaces and help interpreting pattern transitions. In addition, we directly foster the comparison between subspaces by a linked representation and a systematization of occurring pattern transitions.

3 Basic Idea of Pattern Trails

Pattern Trails is a visual interactive approach for expert users, enabling the exploration and understanding of patterns across subspaces of multivariate data. The main idea is to order a set of subspaces in meaningful sequences, such that groups of patterns can be distinguished and their changes effectively traced and interpreted across the subspaces. Pattern Trails follows a three-step procedure depicted in Figure 1 (a). First, we derive interesting subspaces of the multivariate data using a state-of-the-art method for feature selection called SURFING [5]. SURFING searches for subspaces in which data objects form a (hierarchical) clustering structure. The algorithm measures the interestingness of a subspace based on the variance among the k^{th} nearest neighbor of every object. To make the subspaces visually accessible to the user, we then apply DR and project each subspace to 2D space. The results are visualized as scatterplots and arranged side by side in a small-multiple environment [52]. Tatu et al. [51], for example, also visualize all subspace projections, but compare them in a 2D MDS layout without tracing the pattern changes across the subspaces.

Second, we group the subspaces based on their similarity. Subspace search algorithms like SURFING often yield subspaces similar in terms of involved dimensions and/or data relationships, hence producing redundant results. We enable the user to group the projections based on a data-driven similarity measure interactively. Grouping in our experience is essential in subspace analysis, to reduce the abundance of subspaces to a smaller number of representative ones for practical exploration. The projections are also reordered using the data-driven similarity measure which allows the user to set a threshold regarding the level of aggregation.

Finally, we highlight the change of a pattern based on a user-defined selection, which can also be applied automatically using subspace clustering methods. We connect the data points that belong to a selected pattern across all subspaces using lines – we refer to this connected view as pattern *transitions* between subspaces. The transitions describe the structural change of a pattern across subspaces. Changes in subspaces can be modeled as operations to insert, delete or replace subspace dimensions. For example, consider a dense cluster of points spreading over different clusters in the subspace succeeding it in the determined order. This is a significant change in the pattern structure which needs to be further examined in terms of the change of the respective subspace dimensions. This representation may remind of the highly interactive PCPs [22]. However, our approach represents further development regarding the comparison of subspace patterns; it enables to draw conclusions based on links between subspaces of multiple attributes, instead of single attribute axes. Using Pattern Trails, one can find and explain pattern transitions in multivariate data.

Compared to automatic approaches for feature selection (for example, optimizing classification accuracy), Pattern Trails does not search for an optimal set of attributes that form a pattern but rather analyzes which attributes cause a change in the structure of a pattern. Pattern Trails enables the user to analyze any selected pattern in one subspace projection using visual interaction and is not limited by required class-labels or given cluster structures.

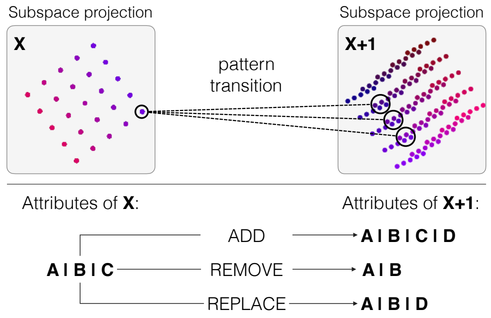


Figure 2: The attributes impact the type of pattern transition which occurs between two subspaces. A transition always migrates from one subspace to the subsequent. In this image, a cluster splits up into three clusters, caused by the operation performed in attribute space coming from the attributes **A**, **B**, and **C**. A pattern alters or remains unchanged among subspaces based on (1) *adding*, (2) *removing*, or (3) *replacing* attributes that build the subsequent subspace. Each operation can be interpreted differently regarding the transition type.

4 Pattern Transitions and their Interpretation

Our Patter Trails approach highlights the change of patterns in multivariate subspaces. As pattern change, we refer to clusters, outliers, or correlations that vary *structurally* across the projections of different subspaces. To make these structural changes visible and accessible to the user, we project the data by means of DR into the two-dimensional space, and represent the results as a scatterplot. An example is illustrated in Figure 2. In the first projection X, the pattern corresponds to a cluster that divides into three clusters in the

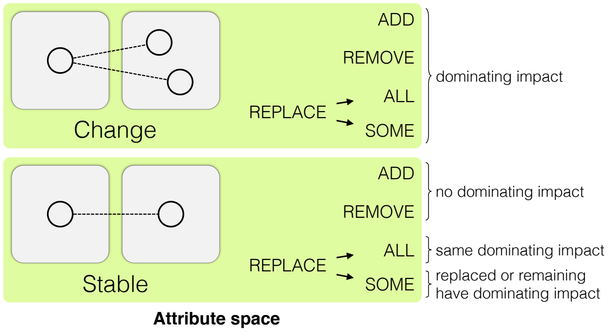


Figure 3: Overview of add, remove and replace operations in the attribute space. Pattern transition between two adjacent subspace projections can show one of two different states: changing or stable. Depending on the operation in the attribute space, the affected attributes imply a different meaning. If a pattern changes, the affected attributes, which contribute to the subsequent subspace, dominate the change. For a stable pattern transition, the affected attributes have either no dominating impact or the same dominating impact as the attributes of the preceding subspace.

second projection $X+1$. The transition is visualized by connecting lines. We conceive there exist two fundamental transition types, as depicted in Figure 3: a *changing* and a *stable* transition, leading to the question: *What is the meaning of these transitions?* To enable the interpretation of such transitions, it is of high importance to consider the attribute space, because it provides information about which attributes are dominant and influential to the structure change of subspace patterns, and which have less influence.

We can distinguish between three basic operations which lead to an either changing or stable transition: Attribute(s) can be *added*, *removed*, or *replaced*. Each operation impacts the interpretation of patterns with respect to the dominance of the attribute(s). An attribute is considered as *dominant* if it significantly controls the structure of the subspace, non-dominant and probably redundant, otherwise. As a result of adding, removing, or replacing attributes, a pattern alters or remains unchanged in the subsequent subspace. The combination of domain-specific knowledge and the visual representation is paramount to explain pattern transitions. Figure 2 and Figure 3 provide an overview of operations and interpretation regarding the attribute space. Note that the geometry of the transition enables us to identify the transition type and the topology enables us to interpret it in terms of the attribute space. Based on the distinction between transition states (changing or stable), we next derive a taxonomy of occurring pattern transitions. The combination of all transitions among all subspaces builds the pattern trail. Following, we discuss the interpretation of pattern transitions with regard to the attribute space and the transition classes (see Figure 4).

4.1 Single Point or Cluster Pattern

The first class of transition patterns refers to single point/cluster transitions.

Static Single Pattern (P1): Within the transition to another subspace, a single point or cluster remains in its consistent/static state. In the case of a single point, the point does not become a member of another pattern, and in the case of a cluster, the cluster does not split or merge with other clusters. The interpretation concerning the attribute space is as follows: *Added* or *removed* attributes are non-dominant and have no impact on the subspace structure. *Replaced* attributes, however, can be dominant if the pattern remains stable. But it is also possible that the untouched attributes are the dominant ones, which impacts the structure.

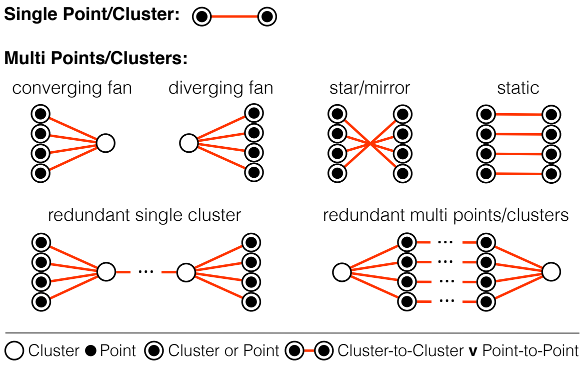


Figure 4: Taxonomy of pattern transitions grouped into transitions between single and transitions between multiple points/clusters. The icons are a conceptualization and represent a simplified version of the real connected subspace views; this means many cluttered lines connect clusters in neighboring subspace views, similar to PCPs.

4.2 Multi Points or Clusters Pattern

In the second class of pattern transitions, we consider all transitions that involve more than one point or cluster. Furthermore, we introduce pattern developments among several subspaces leading to the identification of possibly redundant attributes and subspaces.

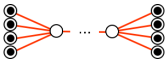
Converging Fan Pattern (P2): Points and/or clusters merge into a single cluster. This transition type indicates a major change by means of the subspace structure. As a matter of fact, the subsequent subspace contains information that reinforces the similarity between patterns and causes them to merge. That is, in attribute space *added* attributes contain information that reinforces the similarity and thus dominate the creation of a cluster. *Removed* attributes take away information. The remaining attributes share more similar information causing the points/clusters to merge. *Replaced* attributes can be interpreted as either dominating or non-dominating, depending on whether they take off or add information causing the patterns to merge.

Diverging Fan Pattern (P3): The diverging fan corresponds to the inverted converging fan **P2**. This means, information is added or removed causing a cluster to split; similar patters reorganize in different groups. *Added* and *Replaced* attributes dominate the subsequent subspace structure and append information so that the cluster content regroups. *Removed* attributes take away information that caused the formation of a cluster at first. Without this information the overall similarity within the cluster decreases.

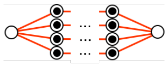
Static Pattern (P4): Similar to **P1**, the static pattern describes a stable transition without changes between sets of patterns. *Added* or *removed* attributes are non-dominant and have no impact on the subspace structure. *Replaced* attributes can be either dominant or non-dominant, depending on whether the untouched attributes dominate the subspace structure.

Star/Mirror Pattern (P5): The interpretation of this transition pattern is identical to the static pattern **P4**. The effect of mirroring can be traced back to the creation of the two-dimensional plots. They are created using a planar projection strategy, such as MDS or PCA, that are known for not being mirror/rotation invariant. Whenever this pattern transition appears, the projection technique mirrored the underlying data.

Understanding the impact of attributes is key to interpret the subspace structure as well as the meaning of a pattern. Consider one pattern (e.g., cluster or outlier) that occurs in an arbitrary subspace. The attribute space provides details in three general scenarios: First, the same pattern occurs in various subspaces. Second, the pattern only occurs in one subspace. Third, the pattern occurs in different structures in various subspaces. For each scenario, we need to draw conclusions in the attribute space to determine dominating, non-dominating, and redundant attributes. This way, we can determine expressive attributes that steer the structure of subspaces. The following two patterns describe combinations of pattern transitions among several subspaces and open the space for attributes that can be considered as being redundant; that is without significant impact to the subspace structure.



Redundant Single Cluster (P6): First, points/clusters converge to a single cluster. Then, the single cluster remains stable among subspaces and finally diverges again to different points/clusters which are not necessarily identical to the initial ones. The interpretation is as follows: The attributes causing patterns to merge and split impact the subspace structure so that patterns regroup. Since these attributes have an impact on the structure, they are likely to hold information that is interesting for further analysis. For all transitions between the subspaces 2^{nd} and $(n-1)^{th}$, the added, removed, or replaced attributes do not show specific impact on the structure of the subspace and the formation of patterns. Therefore, we consider these attributes as being *redundant* in terms of their impact or expressiveness.



Redundant Multi Points/Clusters (P7): This pattern represents the inverse situation to **P6**. First, a cluster diverges. Then, multiple points/clusters remain stable among several subspaces, before they finally converge to a single cluster. As for **P6**, added, removed, or replaced attributes that do not show specific impact on the subspace structure between subspaces $2nd$ and $(n-1)^{th}$, can be considered as being redundant. The information they take away or bring in is not expressive enough to cause the structure to change.

Different pattern transitions can occur within one transition between two subspaces. However, it becomes challenging to interpret the visual depiction. For example, an attribute can be dominant for one pattern transition but redundant for another one such as the combination of the patterns **P1** and **P2** within one single subspace transition. Furthermore, many combinations are possible.

4.3 Automated Support for Interpreting Patterns

One major challenge of Visual Analytics is the automatic support of users in interpreting patterns [43]. In our approach, we consider transitions between subspaces and aim to identify structural changes in patterns based on operations in the attribute space. This leads us to the question: *How can a visual analysis system support the user in understanding pattern transitions?*

Based on the two abstract transition types, depicted in Figure 3, we are able to provide interpretation aid. Generally speaking, if a pattern changes with the transition, the affected attributes dominate the subsequent subspace structure, whether they are added, removed, or replaced. This is different for a stable pattern. Either the affected attributes have no dominating impact (add, remove) or they may have the same dominating impact as the attributes of the preceding subspace (replace). This gives us a powerful tool. In combination with the automatic detection of pattern transitions (see Section 6.3), we can suggest a valid interpretation. Even for combinations of transitions, we can provide a compound of possible interpretations, yet, it is up to the user to employ this information and to gain new insight.

This approach is applicable for verifying hypotheses about the data, and also for explorative tasks such as identifying interesting subgroups and changes. However, the identification and interpretation of a pattern transition depend directly on a meaningful ordering of the subspace representations. In the next Section 5, we discuss the similarity-based ordering of the subspace representations.

5 Similarity-based ordering of Subspace Views

We make use of distance-preserving projections as a means to visualize subspaces of multivariate data. A visual representation makes subspaces accessible and enables the efficient identification of patterns, such as clusters and outliers. To understand how a pattern evolves among different subspaces, we consider their pattern transitions. While a pattern transition is unique between a pair of subspaces, it is challenging to find relevant transitions beyond multiple subspaces, in particular using a visual representation. Simply lining up all subspaces one after another raises the question for a meaningful ordering, which enables us to efficiently identify relevant pattern transitions, so that we can draw conclusions regarding relevant and redundant attributes (consider, e.g., the patterns **P6** and **P7**). A meaningful ordering for multiple subspaces is key for this task.

The issue of finding a meaningful ordering is a known NP-Hard problem and well-known in the parallel coordinate plots [22] domain. It is still subject of ongoing research on how to re-order the axes to obtain expressive results. Examples of ordering goals include an ordering based on maximum pairwise correlations or image-based metrics like reducing the number of line crossings [13]. The main problem of finding a meaningful ordering is that each axis has at most two neighboring axes, as it is the case for our representation; each subspace can be visually connected to at most two neighboring subspaces. However, the problem is different. In contrast to parallel coordinate plots, we handle transitions between two-dimensional projections of multivariate data that are built by more than one attribute. To find an optimal predecessor and successor, we like to consider the notion of similarity between subspaces, in particular, the similarity between their visual representations. This is due to the visual representation of transition types that enables us to interpret on which level patterns are similar or different among subspaces. Expressing the similarity between subspaces is possible in many ways, but two natural ones are based on the similarity between the attribute sets that make up the subspaces, as well as based on the visual similarity between subspaces. We discuss in this Section the application of an attribute-based similarity and introduced a new similarity measure based on the multivariate projection.

5.1 Attribute-based Similarity

It seems apparent that the similarity between two subspaces can be expressed by the similarity of the attribute sets. Two prominent examples used by state-of-the-art approaches are the Jaccard similarity [23] and the Szymkiewicz-Simpson [49] coefficient (compare, e.g., Tatu et al. [51] and Hund et al. [20]). Both coefficients are based on the set of attributes rather than the subspace structure. While the Jaccard similarity provides a ratio of common attributes, the Szymkiewicz-Simpson coefficient considers the exact amount of overlapping attributes among subspaces. However, the attribute-based similarity poses a suboptimal solution, which is why we omit its application in the context of Pattern Trails. Just because the same attributes are used to some extent does not provide any information about the similarity of the data and the projection.

5.2 Projection-based Similarity

A common approach to compute the similarity between pairs of subspace projections is to apply image-based similarity measures to the visual depiction of each projection. For example, Lehmann and Theisel [34] investigate the affine transformations between pairs of projections with the goal to find the most expressive, discriminative projections. However, projections transform the data to a lower-dimensional space and thus inevitably introduce a bias, the projec-

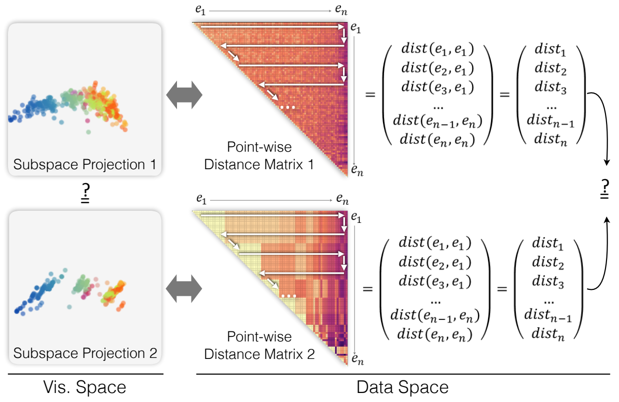

e_1
e_n
e_1
e_n
Subspace Projection 1
Point-wise Distance Matrix 1
$= \begin{pmatrix} dist(e_1,e_1) \\ dist(e_2,e_1) \\ dist(e_3,e_1) \\ \dots \\ dist(e_{n-1},e_n) \\ dist(e_n,e_n) \end{pmatrix} = \begin{pmatrix} dist_1 \\ dist_2 \\ dist_3 \\ \dots \\ dist_{n-1} \\ dist_n \end{pmatrix}$
$\overset{?}{=}$
$\overset{?}{=}$
e_1
e_n
e_1
e_n
Subspace Projection 2
Point-wise Distance Matrix 2
$= \begin{pmatrix} dist(e_1,e_1) \\ dist(e_2,e_1) \\ dist(e_3,e_1) \\ \dots \\ dist(e_{n-1},e_n) \\ dist(e_n,e_n) \end{pmatrix} = \begin{pmatrix} dist_1 \\ dist_2 \\ dist_3 \\ \dots \\ dist_{n-1} \\ dist_n \end{pmatrix}$
Vis. Space
Data Space

Figure 5: Similarity computation between two subspace projections. A possible solution to the question *how to compute the similarity between two subspace projections?* using the example of MDS. We transform the point-wise distance matrices (input for the subspace projections) to 1D feature vectors. Compared to the projection results, the matrix is invariant to rotation, scaling, and translation. We derive the similarity based on standard distance measures between vectors.

tion error, interfering with the notion of similarity. Furthermore, projections are not invariant to rotation, scaling, and translation.

In this section, we provide a solution to the question: *How to compute the similarity between two visual projections of subspaces, namely the scatterplots?* Even though there exist approaches to overcome the named issues, we aim for a solution that is invariant to rotation, scaling, and translation. To do so, we look at how a data projection is computed. In our approach, we apply MDS to the data in order to derive a visual representation of the underlying subspace. Thereby, the MDS derives the final layout by computing a distance matrix and then preserving the distances in a two-dimensional manner. The commonality between projection techniques such as MDS [12], PCA [26], or t-SNE [38] among others is, that they derive the final layout based on an input matrix: either a distance matrix, a covariance matrix, or a probability distribution matrix. We propose to consider the matrices rather than the visual representation, because no projection error is yet introduced, and matrices are known to be invariant to rotation, scaling, and translation. Also, the ordering of rows and columns is not of major concern when computing distances.

We elaborate the question of how to compute the similarity between two projections of subspaces in Figure 5 with application to MDS. Each depiction of a subspace projection is based on a distance matrix that consists of the aggregated distances between pairs of data records. In order to linearize the matrix, we traverse the matrix row-by-row. This way we build an n-dimensional feature vector, whereas n describes the number of data records. Based on the feature vectors, we can compute the distance between two projections using distance functions like Euclidean or Manhattan distance. Visually, our similarity approach orders the subspace projections in terms of the spread between data points in the projection, which is due to the content of the distance matrices. Consider two distance matrices with very different data variances. Naturally, the distance between both matrices is significantly large, which is also reflected by the visual representation. This distance computation is based on the data rather than the sets of contributing attributes and overcomes projection errors introduced by the visual representation. However, the visual representation can still suffer from the projection error. As a result, subspaces may be very similar, but their visual representation significantly differs. One way to overcome this issue is to validate identified patterns with different projection or complementary analysis techniques, which we leave for future work.

Ordering Computation. Based on the derived similarity, we compute all pairwise distances between subspaces to determine an ordering. This means we compute a distance matrix of distances/similarities of all pairwise subspaces. The distance matrix can serve as input to any technique that linearizes the distances, or in other words, preserves the proximities between subspaces in a linear manner. This way, we cannot only provide an ordering of subspaces but also visually point out how close subspaces are to each other. This ordering is special in a sense that it is interpretable in both directions: from left to right and from right to left. The most dissimilar subspaces are located at opposite sites. The closer the subspaces move, the more similar they are. We present results for the well-known iris data set [35] in Figure 6. In this depiction, we compare the (2) Jaccard similarity and the (3) data-driven similarity based on the input matrix. In comparison, the attribute-driven (2) Jaccard similarity performs worse because it ignores the underlying data. The (3) similarity based on the input matrices clearly separates the *Petal Width* and *Height*, which are known for steering the clusters in this data set. The bottom row shows the results after applying user-steered Agglomerative clustering between subspaces. For each cluster, the projections are replaced with a new projection taking into account all clustered attributes. An interesting observation is that the similarity between projections also reflects the spread in the data. From left to right, the point clusters move closer while the transitions remain static. In this example, no prior subspace analysis is applied, yet we can find relevant subspaces and explain their meaning regarding the similarity and the contributing attributes. The displayed pattern transitions in the bottom row furthermore reveal a static development (pattern **P4**) suggesting that in combination with the similarity ordering, the attributes *Petal Width* and *Height* control the subspace structure.

6 Visual Identification of Patterns

A key task in multivariate data analysis is the identification of relevant subspaces and patterns. Patterns, however, can change their overall structure among subspaces, and thus express a different meaning. To explore subspaces and the pattern transitions, we introduce a visual approach based on Sections 4 and 5. The general goal is to integrate the categorization, as well as the similarity-based ordering and clustering of subspaces. Our visual approach comprises a horizontal perspective on the objects and a vertical perspective on the contributing attributes. That is, the visualization of the subspace projections in a horizontal manner side-by-side, and below each projection (vertically), an overview of contributing attributes (compare to the visualization of projections in Figure 6). Thereupon, we employ interaction as means to explore pattern transitions.

6.1 Subspace Cube

Pattern transitions represent an interpretable structure imposed to order and compare patterns between subspaces. The basic idea is to visualize subspaces side-by-side, enabling the identification of changes between adjacent patterns. To visualize the subspaces, we project each subspace to two-dimensional space using the distance-reserving projection MDS [12]. A 3D cube, similar to space-time cubes [4], is a clear choice for visualizing pattern changes across 2D subspace projections, leading to the name of this representation: the *Subspace Cube*. Figure 1 depicts the application to the 2012 University Ranking data set. From a visualization point of view, Pattern Trails looks similar to PCPs. Johansson et al. [25] argue that a 2D representation of PCPs is more effective than a 3D representation, which is why we transform the Subspace Cube to 2D in such way that the main structural pattern changes are preserved. Tufte [52], therefore, introduced Small Multiples as an efficient way to visualize discrete changes in the data. Using small multiples, we depict the side top view of the cube by rotating the small multiples by 0.5π. These two views on the cube differ the most from one another, thus preserving the main structural changes.

Compared to the approaches of Fanea et al. [16] or Yu et al. [55],

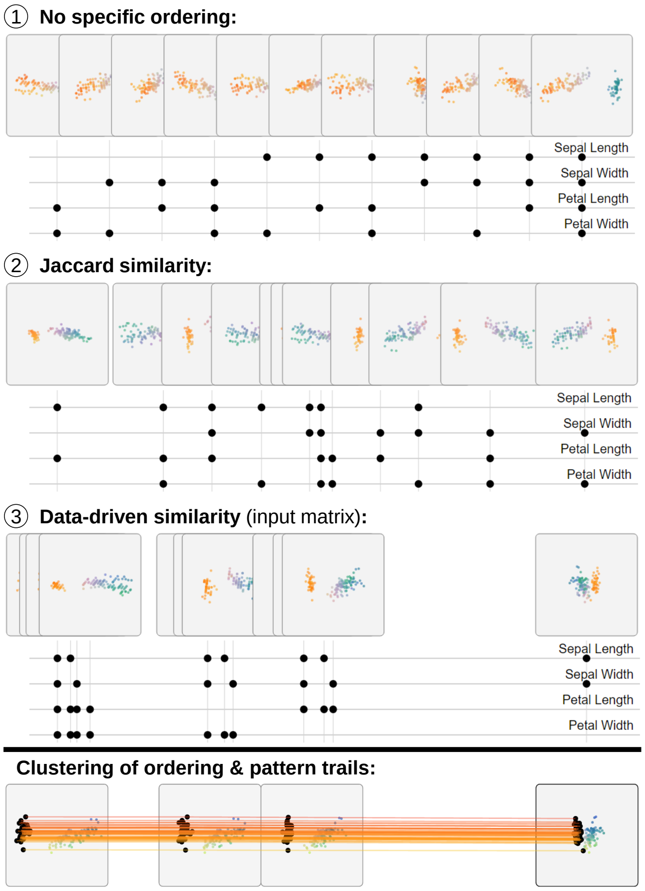


Figure 6: Ordering of the iris data [35]. (1) All subspaces without specific ordering. (2) Application of the Jaccard similarity. (3) Application of our data-driven similarity for projections based on the input matrix. Distances between projections encode similarity, which are linearized using MDS. Very similar projections overlap. *Petal Length* and *Petal Width* steer the clustering and are clearly separated to the rest, compared to the Jaccard similarity. For each ordering the color encoding is re-computed in respect of the first projection. To reduce overlap, we cluster the projections and select a pattern in the visualization, resulting in a static set of pattern transitions.

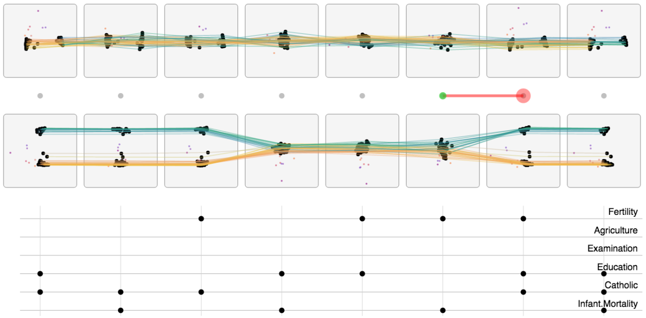


Figure 7: Visualization of the Standardized fertility measure and socio-economic indicators for French-speaking provinces of Switzerland at about 1888 after the application of the subspace analysis algorithm SURFING [5]. The data consists of 6 attributes (*Fertility, Agriculture, Examination, Education, Catholic, Infant.Mortality*), of which the combinations of Fertility, Education, and Infant.Mortality cause the patterns to merge. The pattern is static for multiple clusters with adding the attribute Catholic. This attribute is numeric, but the content is discrete, which causes the pattern to split into two separate clusters: People who are catholic, and people who are not. If this attribute is not considered, then the pattern merges (compare Pattern **P6**). Top: static pattern development. Bottom: view rotated by 0.5π, we see converging, diverging and static patterns. The black points indicate the selection and the lines are colored with regarding the point colors. The diverging fan pattern is highlighted via a red-green connector.

the Subspace Cube visualizes projections and pattern transitions disregarding advanced interaction concepts. Rotating, as well as zooming and panning, are the only available interactions in the hope that the Subspace Cube provides data contexts, which are lost using the small multiple representation. For example, the Subspace Cube provides different angles, which are not available in the small-multiple representation. The projections are equally ordered in the cube as well as the small multiple representation.

6.2 Linked Multiple View

We combine two visual methods to point out the structural change of a pattern across subspaces in the small-multiple environment. The first method color-encodes the projected data points based on a 2D colormap. The idea is to lay all projected points out on a predefined 2D color plane and assign each point the color with identical $x-$ and $y-$coordinates [48]. To point out pattern transitions, we color-encode the data of the first subspace projection and reuse this encoding for all succeeding projections. If the colors mix in a projection, one can draw conclusions regarding the similarity between points – different points are considered to be more or less similar than before.

Color-encoding, however, introduces problems regarding the identification of certain pattern transitions as introduced in Figure 4. For example, it is challenging to differentiate between pattern transitions if many points overlay. Therefore, we additionally visualize the transition between adjacent subspace projections via lines; same points are connected in adjacent projections by single lines. Figure 7 depicts the transitions from a static to a converging, and from a static to a diverging pattern, which is challenging to identify without visual links. The Figure 7 visualizes an additional small multiple row of projections. In the top row the converging and diverging patterns are not visible. Rotating all projections by 0.5π enables the identification of such patterns, which is why we provide both views.

The representation using lines follows the work of Dasgupta and Kosara [13], in which they classified lined-based patterns between axis in parallel coordinates [22]. The interpretation of patterns is different for transitions between adjacent projections, because each subspace projection consists of multiple attributes and is visualized in a way that pattern translations, or mirror images, do not affect the structure of a pattern. This is different for parallel coordinates, where each axis reflects exactly one attribute. Yuan et al. [56] developed a visually similar approach to Pattern Trails which also enables the comparison of multiple attributes. The authors propose to seamlessly integrate scatterplots into parallel coordinates. Thereby, the scatterplot reflects the relations between manually selected attributes (axes) thus allowing a visual inspection of complex patterns. This approach, however, requires the manual selection of subspaces and presents a continuous switch between the PCP and scatterplot representation, posing a challenge to find interesting subspaces and patterns. Pattern Trails relies on small multiples that enable the effective comparison of patterns across many subspaces at a glance.

Similarity-based Ordering and Clustering. The effectiveness of our small multiple representation is highly influenced by a meaningful ordering to make sense of patterns efficiently. In addition, small multiples are limited by the horizontal display dimensions, causing inevitable overlap between multiples, like e.g. depicted in Figure 6. Therefore, we order the small multiples in terms of visual similarity and apply a user-steered Agglomerative clustering: the user is given control over the distance parameter and can interactively change the size of clusters.

In accordance with the ordering computation described in Section 5.2, we use the input matrices of all subspace projections and compute all pairwise similarities. Using a 1D MDS [12], we horizontally reassign the positions of all projections. Distance indicates how similar or dissimilar respective projections are. To reduce overplotting and visual redundancy between projections, and thus subspaces, we apply clustering. Agglomerative clustering has the advantage of only one operating parameter, which reflects the notion of distance between clusters. The user can interactively cluster overlapping subspace projections by steering the distance parameter. A prototype representation then replaces all clusters, this is a new projection including all attributes of the cluster.

6.3 Supporting Pattern Detection

So far, pattern transitions are visualized and found interactively. That is, the user first selects an interesting projection from the small multiples. Then she selects a pattern in the selected projection, for which the full set of transitions is displayed across all subspace projections. Identifying interesting subspaces and patterns is challenging for increasing amounts of data and thus subspaces. We provide *automated support* for finding interesting patterns and transitions by narrowing the representation down to pairwise transitions between adjacent subspace projections. We provide an overview of possible transition patterns in Section 4, Figure 4. The general idea is to point the user to these transition occurrences between subspaces as a starting point for further exploration.

The automatic identification of possibly interesting transitions is based on three criteria: the transition type, the number of affected points that form the transition type, and the minimum distance between affected points. Based on the taxonomy in Figure 4, we distinguish between five transition types. For each type, we automatically detect its occurrence between adjacent pairs of subspaces using *heuristics* based on the Density-based spatial clustering of applications with noise (DBSCAN) algorithm [15]. The core understanding of DBSCAN suits the exploration workflow because it depends on two parameters: the minimum amount of points, and a minimum distance between points that form a cluster. The user can interactively set the size and density of the expected patterns. Based on the user input, we compute the clusters for both subspace projections (following, let us call them projections A and B) separately, and identify the transition type as follows:

Single Point/Cluster: We iterate all clusters in A and check whether a significant amount of points remain in the same cluster in B.

Converging Fan: We iterate all clusters in B and check whether a significant amount of points end up in at least two different clusters in A.

Diverging Fan: Similar to the *Converging Fan*, we reverse the direction and check whether the points emerge from A to B.

Star/Mirror: We iterate all clusters in A and check if a significant amount of points remain in the same cluster. Then, we check if the cluster centroids are mirrored concerning their $y-$ order.

Static: We iterate all clusters in A and check whether a significant amount of points remain in the same cluster in B. Then we check if the clusters are not mirrored.

Based on our experiments, we consider a significant amount of points as 95%. If the relevant condition is met, the transition between two subspaces is detected and highlighted. We include the visualization of visual cues to detected pattern changes, for example, in Figure 1 and Figure 7: A red line with a green starting and a red ending circle indicate an interesting transition. Clicking on this line opens a detail view containing only the affected subspace projections in a scrollable list with all transition occurrences. Our approach does not yet support the identification of multiple transitions between two subspaces, which we leave to future work.

7 Use Cases

We demonstrate the usefulness of Pattern Trails by application to two real-world data sets. We first apply our approach to a University Ranking data set, and discuss our findings in view of automatic subspace analysis applied prior to the visual analysis. Furthermore, we apply Pattern Trails to the well-known Forest Fires data set [35] and report on the analysis workflow, as well as on gathered findings.

7.1 University Rankings

We visually analyze the 2012 World University Rankings [1] data, which suits the general idea of Pattern Trails and comprises seven numerical attributes (scores): *Overall, Academic Reputation, Employer Reputation, Faculty Student Ratio, International Faculty, International Students, Citations per Faculty*. The data was studied by Gratzl et al. [17], who presented a comprehensive system to rank the data including customized preferences. In this work we seek for interesting attributes in terms of subspaces of the data. One can investigate interesting attribute combinations (i.e. subspaces that contain salient patterns) before applying statistic-driven analysis of patterns.

Following, we first examine the data without the use of automatic subspace methods. Then, we perform the analysis again, using an automatic approach before the visual analysis. We show that we come to the same conclusions. A resulting strength of our approach is that it can also be applied without prior automatic subspace analysis.

Manual Analysis of Pattern Transitions

In manual analysis, we load the data, calculate all possible combinations of subspaces, and project them into the 2D space. The data comprises seven attributes. This means, we have to compute 120 (calculation: $(2^n - 1) - n$, with n attributes) combinations, if we ignore all subspaces that consist of exactly one attribute. We then apply our data-driven similarity measure, which is based on the input matrices of the projection. To reduce the 120 subspace projections, we apply agglomerative clustering in addition to the similarity calculation. The result is depicted in Figure 1. A known side effect of clustering is the inherent, continuous information loss because subspaces are combined and replaced by a prototype representation to reduce overlap, and thus cognitive load. Figure 1 points out an example: the same attribute combinations are projected multiple times. Although the subspaces are different in terms of the attribute space, they can result in the same sets of attributes when clustering. However, the visual representation indicates that these prototype representations are dissimilar to each other asking for looking into detail. Clustering is a trade-off between massive overlap at full detail and partial information loss. Interaction is key to overcome this limitation. We enable the user to go back and forth during clustering to alter between overview and detail. Also, one can apply subspace analysis to reduce the sheer amount of subspaces as described in the next section.

We start to read the subspaces depicted in Figure 1 from right to left because the subspaces on the right are clearly separated based on their similarity. The last two subspaces differ by the attribute *Overall Score* and show a static pattern (**P4**). This means, this attribute has no impact on the pattern structure. This behavior repeats between the subspaces 10 and 11, thus not adding new insight. Note that the same attribute sets generate the next three subspaces 8, 9, and 10 and show static (**P4**) or mirror (**P5**) patterns. Again, we cannot reason regarding the attribute space. Between the subspaces 7 and 8 (highlighted in Figure 1), the selected pattern diverges (**P3**) indicating a structural change caused by dominating attributes. To find about which attributes cause this transition, we need to consider the transition between the subspaces 6 and 7. The transition is static (**P4**), meaning that no dominating attribute exists between them. This allows us to compare subspaces 6 and 8 directly. They differ by only one attribute, which is the *International Faculty Score*.

[1] US News/QS 2012 World University Rankings: https://goo.gl/t8zzMe

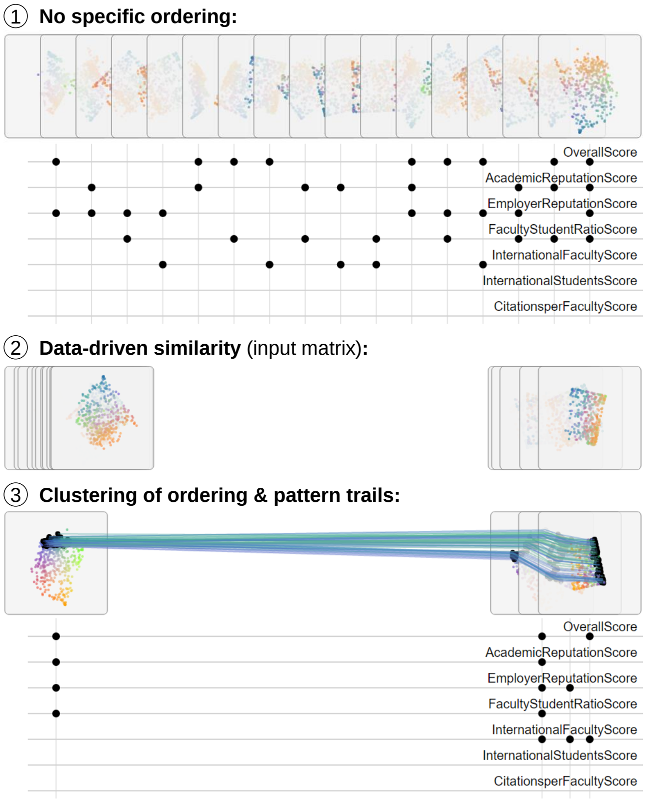


Figure 8: Results of the US News/QS 2012 World University Rankings after application of the SURFING [5] algorithm. (1) The subspace analysis results are shown without any specific ordering. One observation is that the *InternationalStudentsScore* and the *CitationsperFacultyScore* attributes do not contribute to any interesting subspaces. Applying our (2) data-driven similarity measure as well as (3) clustering reveals pattern transitions that show only one split caused by the attribute *InternationalFacultyScore*. The results are consistent with our analysis without the application of prior subspace analysis.

We can consider this attribute as being dominant. Furthermore, the attributes *International Students* and *Citations per Faculty* are removed between the subspaces 6 and 7 and have no impact on the transition, which is why we can consider them as being redundant. The remaining subspaces 1 to 5 show static patterns (**P4**), hence their transitions do not reveal additional insight.

Automatically Aided Analysis of Pattern Transitions

We aim to confirm the results from the previous manual analysis and show how the automatic support helps to improve the analysis. We load the data and then apply the SURFING [5] algorithm. The resulting 16 subspaces are depicted in Figure 8 (1). It is striking that the attributes *International Students* and *Citations per Faculty* play no role, which is in line with the results of the purely manual analysis. (2) We first apply our data-driven similarity and then (3) cluster the results. We select the data points which are separated from the rest by color (blueish color). To reason about dominating attributes, we read the visual representation from left to right. The selected pattern immediately diverges (**P3**) indicating a dominating impact of attributes. However, the first and second subspaces only differ by the attribute *International Faculty Store*. We consider this attribute to cause the pattern transition because the subsequent subspaces are connected by static patterns (**P4**), thus not adding insight. This observation finally confirms the purely manual analysis. Note that we get the same result, but faster, with less effort.

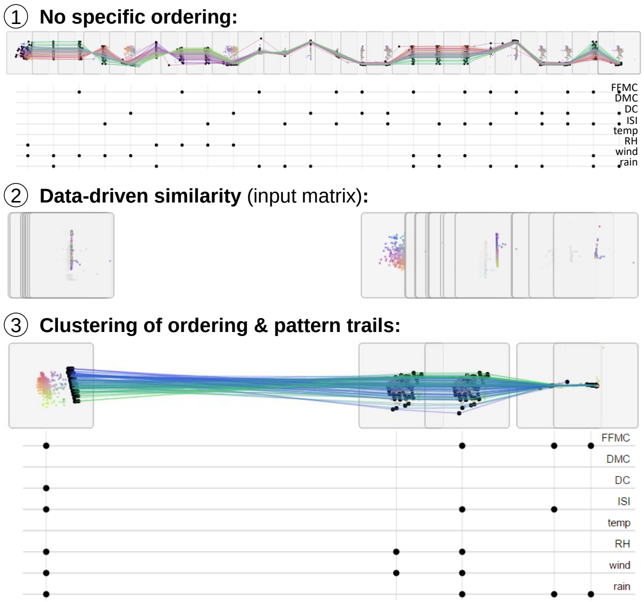


Figure 9: Results of the Forest Fire data set [35] after application of SURFING [5]. (1) Without specific ordering, we identify alternating static and mirror pattern transitions. To expose the role of the attributes in producing patterns, we apply (2) our data-driven similarity and then (3) cluster the subspace projections. The set of pattern transitions reveals a dominating impact of the attributes *Drought Code (DC), Relative Humidity (RH)*, and *wind*.

7.2 Forest Fires

Based on the visual analysis of the Forest Fires [35] data set, we further elaborate the strengths of our approach. The data comprises seven attributes: *Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), Temperature (temp), Relative Humidity (RH), Wind, Rain*. To begin with, we load the data and compute interesting subspaces using the SURFING [5] algorithm. Then, we project and visualize the resulting 24 suggested subspaces, as depicted in Figure 9. We see the subspaces with no specific ordering and the pattern transitions for a cluster in (1), which we selected in the very last projection. We can see several static patterns that develop into each other by means of diverging and converging patterns. We identify 5 interesting transition patterns from left to right. The first 9 subspace projections are combinations of static (**P4**) and mirror (**P5**) patterns, converging (**P2**) into a dense line bundle (**P1**). The converging transition is caused by omitting the attributes *wind, RH*. The line bundle diverges when again considering the attribute *wind*. This behavior is repeated two more times. Also, we can identify the attribute *DC* as being present, except for one time, in the pattern **P1**. Identifying these relations is time-consuming.

To speed this process up, we apply our (2) similarity-based ordering and then (3) cluster the data to avoid artifacts regarding the pattern transitions. The visualization shows that the attributes *DMC* and *temp* do not play a major role because they are not considered at all. Furthermore, the attribute *DC* visually separates the left subspace from the rest. However, we cannot reason about the dominance of this attribute because at no time it causes a pattern to change. Following, we read the transitions from left to right. The first three subspaces are connected by static patterns (**P4**), which do not provide insight regarding the attribute space. The transition between the subspaces 3 and 4 diverges (**P3**), enabling us to reason about dominating attributes. The transition is caused by removing the attributes *RH* and *wind*. The interpretation follows a logical structure because the relative humidity (RH) and the wind most definitely have an effect on the forest fires. The drought code (DC) has most likely the highest impact on the fires. This is held true for the real world, but also for the data.

8 Reflection and Discussion

Pattern Trails is a visual interactive approach that includes the user to identify and explain interesting attributes, and thus subspaces, in possibly large multivariate datasets. Introducing pattern transitions, we enable the user to analyze the structural changes in subspaces and also provide interpretations for these changes. Pattern Trails can be considered as an extension to the statistic-driven analysis of clusters, outliers, and correlations that occur in a subspace. The design space of our approach opens questions regarding different analysis steps that we aim to discuss at this point.

Pattern Selection

Given a number of subspace projections, there exist different strategies to identify a pattern (e.g., a cluster) as a starting point to analyze its transitions. An effective approach is to select all points within a subspace. The data is encoded by a 2D colormap, and the connecting lines take on these colors. Thus, the selection of all data points enables the identification of the main structural changes across subspaces. This approach can be applied to discover interesting pattern transitions and then to reduce the clutter by selecting promising identified data subsets (patterns). The main problem remains the number of subspaces. If subspaces overlay too much, the transitions cannot be kept apart; thus one can not reason about the attribute space. Using our data-driven similarity and applying Agglomerative clustering reduces the number of subspace projections. Aggregation reduces the displayed data, yet at the expense of information. The more attributes are combined, the less expressive the subspace projection can get. It is up to the user to strike the balance between aggregation and aspired information.

Another strategy is to provide automatic support like, for example, the application of data-driven clustering algorithms that point out the salience of patterns in subspaces. However, such algorithms do not point the user to structural changes across subspaces, which is why we go one step further and provide automatic support for identifying meaningful transition pattern.

Attribute Redundancy

The notion of redundant and dominant (or expressive) attributes raise the question for: Should users omit redundant attributes? That may depend. On the one hand, we seek for attributes which determine the structure of the subspaces, which is why we can argue that all other attributes may not sufficiently contribute in terms of information. On the other hand, attributes considered to be possibly redundant can still be classified by domain experts as important with respect to their task at hand. Consider for example 10 attributes, of which 3 attributes determine the structure of the subspace. The remaining 7 attributes are not of major concern regarding the subspace structure, however, contain information that can be relevant to get insight and bring attributes into context. For this reason, we included Pattern Trails into the Visual Analytics process, so that it is up to the user to decide whether certain attributes and subspace structures are of interest.

Expressiveness of Subspace Projections

The visual representation of subspace projections brings in challenges regarding the expressiveness and uniqueness of the subspaces. First, MDS is known to be not invariant to rotation, which is why we introduced the mirror transition patterns. This transition is salient (e.g. in Figure 1), but holds the same interpretation as the static pattern. In addition, we rotated all small multiples to provide an extensive view on the subspace projections and not to leave the user with ambiguities. Second, visualizing the structures in a subspace by a distance-preserving projection is an intuitive way to see the relations between different data points. However, projection techniques introduce biases and do not correctly represent the underlying data [45]. This can significantly influence the interpretation of the perceived patterns. This poses the key challenge of Pattern Trails because the interpretation depends on the patterns generated by the projection. We plan to extend the visual representation of the projection by encoding its quality as suggested, for example, by Martins et al. [40].

We further applied the SURFING [5] algorithm, because it is parameter-free and proposes interesting subspaces based on their structure. We are aware of a variety of other algorithms, but like to leave their integration to future work because they are not an essential part of our claimed contributions.

Subspace Ordering

Using Pattern Trails, we layout all subspace projections side by side and encode similarity between projections via distance. This approach introduces a meaningful ordering in terms of groups of subspaces that are clearly separated. Incorporating different distance functions, however, can introduce diverse orderings that may affect the interpretation of patterns. Although our approach is invariant to orderings, we may come to different conclusions when applying clustering to similar subspace projections. This is due to different groups that comprise different attributes. Therefore, it is of highest interest to confirm findings using state-of-the-art statistical analysis methods or to also investigate attribute combinations without clustering.

Different orderings also introduce possibly arbitrary combinations of transition patterns. We consider the taxonomy presented in Section 4 as complete in terms of single transformations and the identification of redundant attributes, but plan to extend the taxonomy for arbitrary combinations and their interpretation in the future.

Scalability

To assess the computational and visual scalability of Pattern Trails, we consider both, the number of data records n and the number of visualized subspace projections m. For each subspace, we compute the MDS to determine the visual representation, which lies in $\mathcal{O}(n^3)$ due to the eigendecomposition. This computation is parallelizable. The interactive sorting of the subspaces using Agglomerative clustering or the 1D MDS projection adds to the complexity by $\mathcal{O}(m \cdot log\ m)$ or $\mathcal{O}(m^3)$ respectively. From a visual perspective, the scalability of Pattern Trails is highly influenced by the number of displayed subspaces. To optimize them, we propose to apply our data-driven similarity followed by Agglomerative clustering, which is a trade-off between a detailed view on the data and a condensed view on possibly interesting parts.

To tackle a number of data records, we apply a 2D colormap to identify overall structural changes but are aware that this is not an optimal solution. For future work, we imagine including scatterplot-based aggregation techniques such as heatmaps, and bundling techniques for the transition connectors. We applied Pattern Trails to datasets with 8 attributes. However, our technique can also be applied to datasets with a much larger attribute space. This poses a challenge to the computation of interesting subspaces and their visual representation. The number of possible subspaces grows exponentially with the number of attributes. While heuristics such as SURFING determine subspaces with potentially interesting structures, we cannot guarantee that the algorithm keeps all important patterns. Especially in high-dimensional spaces, parameterization and/or interpretation of distances becomes challenging and influences the quality of the analysis results. As for the visual scalability of an enormous amount of subspaces, we refer to the possibility of clustering. We implemented our prototype using a client-server architecture, which allows to handle large amounts of data. However, the computation and results of subspaces and projections are affected by the data characteristics (e.g. amount of attributes), traced to the curse of dimensionality [6], the data types, and the data size.

Acknowledgments

This work was partly supported by the EU project Visual Analytics for Sensemaking in Criminal Intelligence Analysis (VALCRI) under grant number FP7-SEC-2013-608142 and the German Research Foundation (DFG) within project A03 of SFB/Transregio 161.

References

[1] A. Anand, L. Wilkinson, and D. T. Nhon. Visual pattern discovery using random projections. In *2012 IEEE Conference on Visual Analytics Science and Technology, VAST 2012, Seattle, WA, USA, October 14-19, 2012*, pp. 43–52, 2012. doi: 10.1109/VAST.2012.6400490

[2] D. F. S. Andreas Buja, Dianne Cook. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.

[3] D. F. Andrews. Plots of high-dimensional data. *Biometrics*, 28(1):125–136, 1972.

[4] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. In R. Borgo, R. Maciejewski, and I. Viola, eds., *EuroVis - STARs*. The Eurographics Association, 2014. doi: 10.2312/eurovisstar.20141171

[5] C. Baumgartner, C. Plant, K. Kailing, H. Kriegel, and P. Kröger. Subspace selection for clustering high-dimensional data. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*, pp. 11–18. IEEE Computer Society, 2004. doi: 10.1109/ICDM.2004.10112

[6] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[7] S. Bremm, T. v. Landesberger, J. Bernard, and T. Schreck. Assisted descriptor selection based on visual comparative data analysis. *Computer Graphics Forum (Proc. EuroVis 2011)*, 30(3):891–900, 2011. doi: 10.1111/j.1467-8659.2011.01938.x

[8] N. Cao, D. Gotz, J. Sun, and H. Qu. DICON: interactive visual analysis of multidimensional clusters. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2581–2590, 2011. doi: 10.1109/TVCG.2011.188

[9] J. Chambers. *Graphical methods for data analysis*. Chapman & Hall statistics series. Wadsworth International Group, 1983.

[10] S. Cheng and K. Mueller. The data context map: Fusing data and attributes into a unified display. *IEEE Trans. Vis. Comput. Graph.*, 22(1):121–130, 2016. doi: 10.1109/TVCG.2015.2467552

[11] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[12] T. Cox and A. Cox. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2000.

[13] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1017–1026, 2010. doi: 10.1109/TVCG.2010.184

[14] A. Endert. *Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering*. Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2016. doi: 10.2200/S00730ED1V01Y201608VIS007

[15] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, eds., *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pp. 226–231. AAAI Press, 1996.

[16] E. Fanea, M. S. T. Carpendale, and T. Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. In *IEEE Symposium on Information Visualization (InfoVis 2005), 23-25 October 2005, Minneapolis, MN, USA*, pp. 149–156, 2005. doi: 10.1109/INFOVIS.2005.5

[17] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2277–2286, 2013. doi: 10.1109/TVCG.2013.173

[18] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

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

[20] M. Hund, D. Böhm, W. Sturm, M. Sedlmair, T. Schreck, T. Ullrich, D. A. Keim, L. Majnaric, and A. Holzinger. Visual analytics for concept exploration in subspaces of patient groups. *Brain Informatics*, 3(4):233–247, 2016. doi: 10.1007/s40708-016-0043-5

[21] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2010, Salt Lake City, Utah, USA, 24-29 October 2010, part of VisWeek 2010*, pp. 3–10, 2010. doi: 10.1109/VAST.2010.5652392

[22] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In A. E. Kaufman, ed., *Proceedings IEEE Visualization '90, San Francisco, California, USA, October 23-26, 1990.*, pp. 361–378. IEEE Computer Society Press, 1990. doi: 10.1109/VISUAL.1990.146402

[23] P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[24] D. H. Jeong, C. Ziemkiewicz, B. D. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. *Comput. Graph. Forum*, 28(3):767–774, 2009. doi: 10.1111/j.1467-8659.2009.01475.x

[25] J. Johansson, C. Forsell, and M. D. Cooper. On the usability of three-dimensional display in parallel coordinates: Evaluating the efficiency of identifying two-dimensional relationships. *Information Visualization*, 13(1):29–41, 2014. doi: 10.1177/1473871613477091

[26] I. Jolliffe. *Principal component analysis*. Springer series in statistics. Springer-Verlang, 1986.

[27] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph.*, 19(3):495–513, 2013. doi: 10.1109/TVCG.2012.110

[28] D. A. Keim, M. Ankerst, and H. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *IEEE Visualization '95, Proceedings, Atlanta, Georgia, USA, October 29 - November 3, 1995.*, pp. 279–286. IEEE Computer Society Press, 1995. doi: 10.1109/VISUAL.1995.485140

[29] D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets? *Information Visualization*, 1(1):20–34, 2002. doi: 10.1057/palgrave/ivs/9500003

[30] T. Kohonen. Essentials of the self-organizing map. *Neural Netw.*, 37:52–65, Jan. 2013. doi: 10.1016/j.neunet.2012.09.018

[31] J. Krause, A. Dasgupta, J.-D. Fekete, and E. Bertini. SeekAView: an intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. *Large Data Analysis and Visualization (LDAV), IEEE Symposium on*, Oct 2016.

[32] J. Krause, A. Perer, and E. Bertini. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1614–1623, 2014. doi: 10.1109/TVCG.2014.2346482

[33] H. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1), 2009. doi: 10.1145/1497577.1497578

[34] D. J. Lehmann and H. Theisel. Optimal sets of projections of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 22(1):609–618, 2016. doi: 10.1109/TVCG.2015.2467132

[35] M. Lichman. UCI machine learning repository, 2013.

[36] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Trans. Vis. Comput. Graph.*, 23(3):1249–1268, 2017. doi: 10.1109/TVCG.2016.2640960

[37] S. Liu, B. Wang, J. J. Thiagarajan, P. Bremer, and V. Pascucci. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Comput. Graph. Forum*, 34(3):271–280, 2015. doi: 10.1111/cgf.12639

[38] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[39] B. F. Manly. *Multivariate statistical methods: a primer*. CRC Press, 2004.

[40] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, 41:26–42, 2014. doi: 10.1016/j.cag.2014.01.006

[41] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[42] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004. doi: 10.1145/1007730.1007731

[43] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1604–1613, 2014. doi: 10.1109/TVCG.2014.2346481

[44] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Vis. Comput. Graph.*, 23(1):241–250, 2017. doi: 10.1109/TVCG.2016.2598495

[45] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Information Visualization*, 9(3):181–193, 2010. doi: 10.1057/ivs.2010.2

[46] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2634–2643, 2013. doi: 10.1109/TVCG.2013.153

[47] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. Vis. Comput. Graph.*, 22(1):629–638, 2016. doi: 10.1109/TVCG.2015.2467717

[48] M. Steiger, J. Bernard, S. Mittelstädt, M. Hutter, D. Keim, S. Thum, and J. Kohlhammer. Explorative analysis of 2d color maps. In V. Skala, ed., *Proceedings of WSCG*, vol. 23, pp. 151–160. Eurographics Assciation, Vaclav Skala - Union Agency, 2015.

[49] D. Szymkiewicz. *Une contribution statistique a la géographie floristique*. Polskie Towarzystwo Botaniczne, 1934.

[50] A. Tatu, G. Albuquerque, M. Eisemann, P. Bak, H. Theisel, M. A. Magnor, and D. A. Keim. Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 17(5):584–597, 2011. doi: 10.1109/TVCG.2010.242

[51] A. Tatu, F. Maass, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *2012 IEEE Conference on Visual Analytics Science and Technology, VAST 2012, Seattle, WA, USA, October 14-19, 2012*, pp. 63–72, 2012. doi: 10.1109/VAST.2012.6400488

[52] E. R. Tufte. Envisioning information. *Optometry & Vision Science*, 68(4):322–324, 1991.

[53] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions - A dual visual analysis model for high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2591–2599, 2011. doi: 10.1109/TVCG.2011.178

[54] M. O. Ward and A. R. Martin. High dimensional brushing for interactive exploration of multivariate data. In *IEEE Visualization*, p. 271, 1995. doi: 10.1109/VISUAL.1995.485139

[55] B. Yu, R. Liu, and X. Yuan. Mlmd: Multi-layered visualization for multi-dimensional data. *The Eurographics Association*, 5:103–107, 2013.

[56] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1001–1008, 2009. doi: 10.1109/TVCG.2009.179

[57] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2625–2633, 2013. doi: 10.1109/TVCG.2013.150

[58] F. Zhou, J. Li, W. Huang, Y. Zhao, X. Yuan, X. Liang, and Y. Shi. Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In *2016 IEEE Pacific Visualization Symposium, PacificVis 2016, Taipei, Taiwan, April 19-22, 2016*, pp. 128–135, 2016. doi: 10.1109/PACIFICVIS.2016.7465260