Vision, Modeling, and Visualization (2024)
L. Linsen and J. Thies (Eds.)

Not Just Alluvial: Towards a More Comprehensive Visual Analysis of Data Partition Sequences

Madhav Poddar [†1], Jan-Tobias Sohns [‡2], Fabian Beck [§1]

[1]University of Bamberg, Germany
[2]University of Kaiserslautern-Landau, Germany

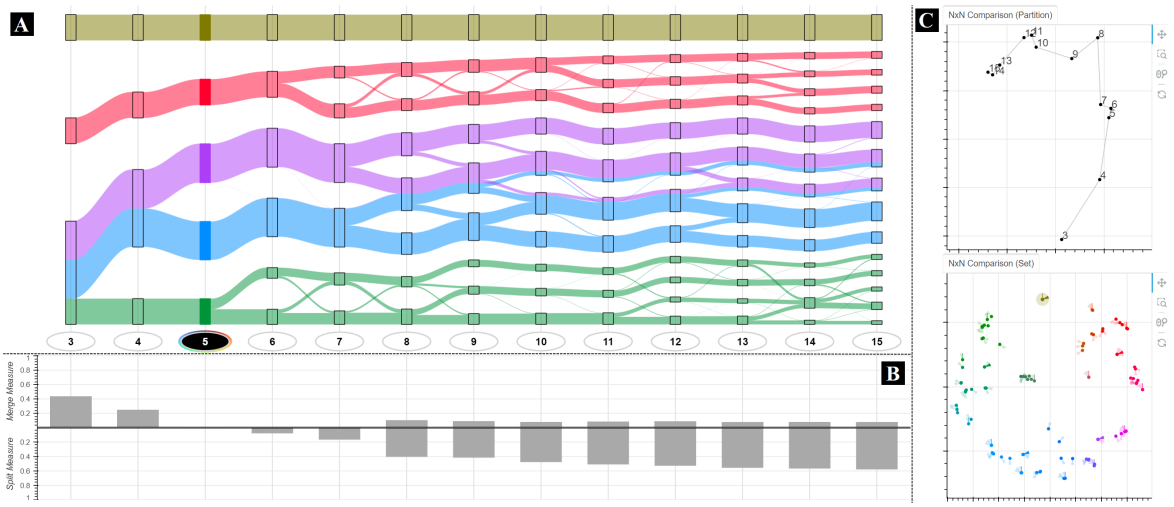

Figure 1: *Enhanced alluvial diagram (A) complemented with component (B) to compare a selected partition with all other partitions regarding splits and merges, and components (C) to provide projected similarities of all partitions and sets, respectively. The figure depicts different clustering results obtained by varying the number of clusters in k-means. Partition 5 is selected and determines the coloring.*

Abstract

Data items arranged into groups form partitions, and across time or through variation of grouping criteria, those partitions may change. While alluvial diagrams, showing the flow of data items as streams, visually capture such changes in partition sequences, their focus on showing similarities between neighboring partitions limits their application. Our paper introduces novel augmentations of alluvial diagrams with interactive visualizations and linked analysis, explicitly targeting the comparison of non-neighboring partitions without sacrificing the sequential nature of the data. Juxtaposed visualizations with the alluvial diagram's timeline provide a comparison of a selected partition to all other partitions, while additional scatterplot views provide an overview of the partition and set similarities. Connecting the set representations across views, we propose a coloring approach of sets and interactive selection mechanisms. The usefulness and generalizability of the approach are demonstrated through examples with application in supervised and unsupervised machine learning, as well as work collaboration analysis.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics; Information visualization;***

1. Introduction

In this paper, we analyze sequences of partitions of data items in different applications. In a *partition*, all considered data items—

† madhav.poddar@uni-bamberg.de
‡ sohns@rptu.de
§ fabian.beck@uni-bamberg.de

DOI: 10.2312/vmv.20241202

called *population* in set theory—are divided into sets. More precisely, it is a division of the items into non-empty sets, ensuring that each item of the population is precisely included in one of these sets. Clustering algorithms, such as k-means, produce partitions automatically from characteristics of the data items, trying to group *similar* items in each set of the partition. However, usually, various valid partitions exist. For instance, in the context of k-means, adjusting the hyperparameter k—the desired number of clusters—produces partitions on different levels of granularity. Organizing these partitions in ascending order of k forms a specific sequence, providing insight into multiple levels of valid groupings of the data items. Alternatively, sequences of partitions might stem from dividing the population at different points in time, according to a given data attribute or by clustering.

To visualize sequences of partitions, *alluvial diagrams* can be used. Bars denote sets within a partition, and edges connect the bars to related sets in the neighboring partitions. The height of the bars and edges reflects set size, contributing to the metaphor of branching and merging streams carrying different quantities of items. Figure 2 provides an example showing the evolving partition of seven data items across three steps. Alluvial diagrams clearly reveal patterns in the sequence of partitions and thereby support a $\mathbf{1} \times \ldots \times \mathbf{1}$ comparison of the partitions. However, key limitations revolve around the diagram's focus on neighboring partitions, which can overshadow relationships between non-neighboring ones. For instance, in Figure 2, despite the small size of the dataset, it takes quite some effort to find, for the green set in the first partition, the most similar one in the third partition (solution: there are two, $\{2,6\}$ and $\{4,0\}$). Supplementary visualization techniques may be required to draw a more comprehensive picture of similarities of partitions and contained sets across the whole sequence, also taking perspectives of $\mathbf{1} \times \boldsymbol{N}$ and $\boldsymbol{N} \times \boldsymbol{N}$ comparisons.

Recognizing these limitations, in this work, we enhance alluvial diagrams by infusing interactive selections and linked views as shown in Figure 1, thereby broadening their capacity to support $\mathbf{1} \times \ldots \times \mathbf{1}$, $\mathbf{1} \times \boldsymbol{N}$, and $\boldsymbol{N} \times \boldsymbol{N}$ comparison modes. First, we optimize the alluvial diagram, at the center in Figure 1(A), by a similarity-based coloring approach, which also supports selecting a partition of interest to determine the colors of the streams. Second, shown in Figure 1(B), a bar chart, juxtaposed below, provides further information on splits and merges, comparing the selected partitions with all other partitions. Third, two scatterplots, displayed in Figure 1(C), explicitly project similarities of partitions and sets to a two-dimensional space, still hinting at their sequential position. Since our approach is not specific to certain applications, we demonstrate its generalizability and value through three application examples from different domains with varying data characteristics. Hence, our main contributions are to introduce and apply a generalizable, interactive, multi-view approach for extending alluvial diagrams towards more comprehensive support for comparative analysis of data partitions.

2. Related Work

The term *alluvial diagram* was coined in 2010 [RB10], but the concepts it builds on are older. Such diagrams link to *Sankey diagrams*—visualizations of branching and merging streams of quantities encoded in the width of streams—, which date back to the 19th century [Min69, KS98]. However, while not necessary in Sankey diagrams, alluvial diagrams organize the different data partitions in sequential states (typically drawn from left to right as columns).

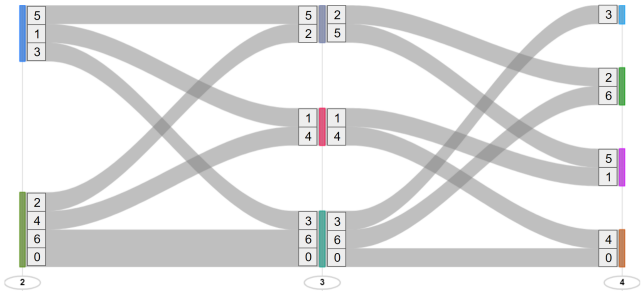

Figure 2: *Alluvial diagram representing a sequence of partitions, here, the remainder of integers 0 to 6 divided by 2, 3, and 4.*

Main Approaches and Applications: The original alluvial diagrams were suggested to visualize evolving clusters in network structures [RB10] and have been extended for this use case in different directions [RTJ*11, VBAW14]. Moreover, similarities to alluvial diagrams can be found in many stream-like, linearly arranged visualizations including, but not limited to *ThemeRiver* [HHWN02] and other topic stream visualizations [LYW*16, BMBW15, CLT*11, XWW*13, CLWW14], *Parallel Sets* [KBH06] and other sequential set visualizations [AB20], different types of aggregated event visualizations [PW14, KABB23], as well as evolving hierarchies [BNRB21, VBW16, TA08, CLWW14]. Application areas span from scientific and social network analysis [RB10, VBAW14, RTJ*11, AB20] to supporting topic extraction [CLT*11, XWW*13, CLWW14] and machine learning [AB20]. Further, storyline representations [TM12, THM15, OM10, TRL*19, LWW*13] are related, which typically visualize the groupings of characters in stories, but can be applied also, for instance, to software evolution [OM10]. While a review of all such visualizations is beyond the scope of this paper, we discuss in the following those works and aspects that specifically link to our suggested extensions of alluvial diagrams.

Alluvial Diagram Layout: Leveraging the metaphor of streams and rivers, alluvial diagrams usually connect the linearly arranged groups in smoothly curved bands or ribbons. Some approaches vary the width of the streams according to quantification of contributions [HHWN02, BMBW15], however, our focus is on qualitative changes and approaches where items have constant width, but streams branch and merge. If elements are not present in all sequential states, they can be indicated as in- and outflows from the top or bottom of the diagram [AB20, BMBW15]. Although we use a constant set of items, such in- and outflows would be a feasible extension. Vertically, the groups can be ordered by a characteristic property (e.g., size [RB10], overlap properties [AB20]). Alternatively, they can be arranged to avoid clutter caused by edge crossings, for instance, based on the Sugiyama layout for hierarchically drawn directed acyclic graphs [VBAW14]. Our solution heuristically computes shortest paths applying set-similarity-based distances, implicitly also avoiding edge crossings.

Enriched Visual Encodings: Additional visual encodings may enrich alluvial and stream diagrams in different regards. Many approaches choose colors based on a set or item property [VBAW14, BZSD21, CLWW14, RTJ*11, TA08, KBH06]. However, the availability of such properties is application-specific. In a more generalizable way, colors can be assigned to sets based on clustering [VBAW14, RB10, ELAS21]. Inspired by these, our solution maps color by set similarity, but more directly using set-based similarities for a projection to the color space. We further borrow the idea of interactively applying color to streams across the sequence when selecting a specific partition [KBH06]. In some applications, it is important to place labels inside the sets, for instance, for topic streams and related visualizations [LYW*16, BMBW15]. Storyline visualization might annotate each stream with a label [OM10, THM15]. If the alluvial diagrams show groups within networks, providing network context is relevant, drawn as links within and between groups or items [VBAW14, ELAS21], or visualized as adjacency matrices in the columns [VBW16]. As we study the general case of unlabeled and unconnected items, we do not integrate any of such labeling or network-based solutions.

Multi-view Approaches: Other views might complement alluvial and stream diagrams to contribute additional perspectives beyond details-on-demand views or selection panels. For instance, Wu et al. [WZW*16] suggest additional cluster and map views along with an alluvial diagram that summarizes urban mobility data. For game analytics, Chen et al. [CLK*17] visualize player status transitions in alluvial diagrams accompanied by different statistical diagrams. Visually summarizing sets of call graphs, Kesavan et al. [KBB*23] link their Sankey diagrams with scatterplot, distribution, and projection views. While this demonstrates the potential of enrichment through additional views, these solutions are application-specific. Our approach makes use of additional visualizations for complementing visual comparison of non-neighboring partitions and sets—we are not aware of another general approach with similar extensions to alluvial diagrams.

3. Visualization Design

In our proposed approach, the alluvial diagram is linked with a $\mathbf{1} \times \boldsymbol{N}$ and two $\boldsymbol{N} \times \boldsymbol{N}$ comparison views. Selections made in the alluvial diagram extend their influence to the associated views. We implemented the proposed approach as a multi-view system in Python using the Bokeh library [Bok18]. The source code [PSB24] is available under an open license, and a video demonstration is included as supplementary material.

3.1. Data Model

Our approach visualizes a sequence of partitions of data items. Formally, let $D = \{d_1, d_2, \ldots, d_n\}$ denote a population of data items. Then, a *partition sequence* of D is formalized as $\mathcal{P} = (P_1, P_2, \ldots, P_N)$ where each $P_i = \{S_{i,1}, S_{i,2}, \ldots, S_{i,k_i}\}$ is a partition of D. This means sets $S_{i,j}$ in each partition P_i are nonempty ($\emptyset \neq S_{i,j} \in P_i$) and disjunct from each other ($S_{i,j}, S_{i,j'} \in P_i, j \neq j' \Rightarrow S_{i,j} \cap S_{i,j'} = \emptyset$), as well as all sets in P_i together represent the whole population ($\bigcup_{S_{i,j} \in P_i} S_{i,j} = D$). Hence, n refers to the number of data items and N to the length of the sequence. Additionally, k_i denotes the number of sets in the i^{th} partition and can be different for each partition in the sequence. Partitions P_i might be assigned labels l_i (or $l_i = i$). Data items or sets do not carry labels.

3.2. Alluvial Diagram

The alluvial diagram serves as the core of our approach and provides the basis for a sequential $\mathbf{1} \times \ldots \times \mathbf{1}$ comparison of partitions and sets. The diagram is made of two visual elements. First, rectangular bars indicate sets $S_{i,j}$, and second, edges each connect two sets in neighboring partitions that have common data items. The height of the bars indicates the cardinality of the set $|S_{i,j}|$. Inherited from the bars, the strength (height) of the edges denotes the cardinality of the intersection of the two sets it connects ($|S_{i,j} \cap S_{i+1,j'}|$). Whereas the alluvial diagrams visualize changes between neighboring partitions, they do not explicitly support comparisons of non-neighboring partitions. Addressing this already partly within the diagram, we extend it and offer two perspectives.

Overview Perspective: Providing an overview is the default perspective of an alluvial diagram—trying to show all relevant information in a single static view. Colors can be used to improve the viewers' ability to follow the set changes and better compare sets from different partitions. Moreover, the vertical ordering of sets and edges is generally relevant to give the sets a meaningful structure and avoid unnecessary edge crossings. As illustrated in Figure 3, we suggest specific solutions to both aspects that prepare and align with the comparison of non-neighboring partitions that the further views support. Figure 6 shows an example of the resulting alluvial diagram in the overview perspective.

Assigning colors to each set is important, as color can serve as a visual cue when interpreting an alluvial diagram. Appropriate color allocation can highlight stable patterns or transitions effectively. Moreover, non-neighboring sets can be compared if colors relate to their similarity. Hence, sets with similar members should carry similar colors, which we implement through the following heuristic approach, exemplified in Figure 3 (top). Based on the *Jaccard similarity coefficient*, we define a distance metric between sets

$$d(S_a, S_b) = 1 - \frac{S_a \cap S_b}{S_a \cup S_b} \quad .$$

By using the distance matrix for all pairs of sets $S_{i,j}$ as input, we run *multi-dimensional scaling* (MDS) [Tor52] to compute a two-dimensional embedding. As far as possible, the embedding places similar sets nearby, and in turn, dissimilar sets farther apart. The embedding space is then mapped to a two-dimensional perception-based CIELAB color space (with constant luminance L). A consequence of this approach is that sets with some common members with most other sets tend to lie in the center of the color space, being assigned low saturation colors (gray or close to it). This is a desirable side effect, giving those in a sense *central* sets always similar color shades, for instance, some larger sets in Figure 6.

The vertical ordering of the sets directly impacts the number of edge crossings, which in turn affects the readability of the alluvial diagram. To address this issue, we determine a sequence of sets based on a heuristic solution [Chr22] of the *traveling salesman problem* on the two-dimensional embedding of the sets, which is

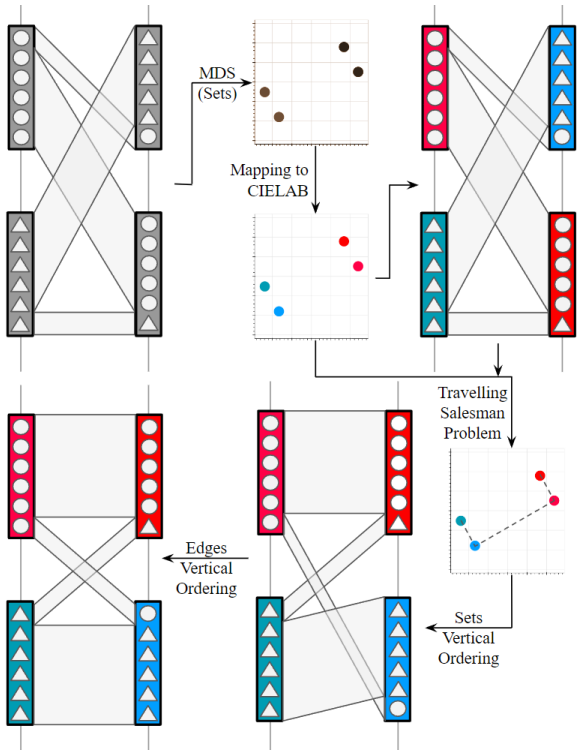

Figure 3: *Illustration of the set-similarity-based algorithm for color assignment and vertical ordering in the alluvial diagram.*

computed for color assignment. This places similar sets closely and thereby helps reduce edge-crossings. Moreover, within the sets, the edges are subsequently ordered vertically to reduce the number of edge crossings further. Between every pair of connected neighboring partitions, we first draw the edges based on the vertical order of sets in the left partition. For edges originating from the same set in the left partition, the edges are drawn based on the vertical order of the sets in the right partition. Figure 3 (bottom) illustrates that this set and edge ordering makes the alluvial diagram less cluttered.

Selected Partition Perspective: While, through colors, the overview perspective already allows identifying similar sets in non-neighboring partitions, it is still difficult to compare partly similar sets and quantify their difference. To address this, users can select a specific partition (column) or sets (bars), and see the changes across the sequence from the perspective of the selected partition or sets.

A click on the partition's label selects it. While the set colors in the selected partition P_s remain unchanged, the ones of the non-selected partitions adapt to reflect the selection. To this end, edges are subdivided into differently colored sub-edges based on the membership of corresponding data items in the selected partition. Sets from the selected partition $S_{s,j}$ keep their appearance, but sets from non-selected partitions become transparent. When sub-edges pass through a set, we group them by color and get a part-to-whole representation of set membership regarding the selected partition. By this assignment of colors based on the selected partition, we allow tracing where the item members of sets $S_{s,j}$ move.

Within this perspective, we added support for filtering based on one or more sets from the same partition to reduce visual complexity and focus on sets of interest. Filtering is triggered by clicking on the bars of the sets and draws only the sub-edges associated with them (as shown in Figure 8). Note that we do not support selection of multiple sets from different partitions to avoid complications, like whether to consider union or intersection between overlapping sets and additional issues concerning the linking of views.

With the division of edges into sub-edges, an additional algorithm is required to avoid sub-edge crossings. As we group the sub-edges based on color (dependent on the selected partition P_s), we cannot keep the edge layout of the overview perspective and just subdivide them into different colors. The sub-edges are computed, and their order needs to be recomputed every time a new partition (or sets within a partition) is selected. We append a sorting key to the previously discussed edge sorting in the overview perspective to do this. For sub-edge sorting, we first place edges based on the vertical order of the selected partition, followed by the order of the left partition, and finally based on the order of the right partition.

3.3. $1 \times N$ Comparison View

To provide more support for $\mathbf{1 \times N}$ comparisons beyond assigning colors and filtering, we add a view to summarize the set membership change across the sequence from the perspective of one specific partition. It aligns with the sequential axis of the alluvial diagram and is placed below as shown in Figure 1(B).

We define a merge measure δ^+ and a split measure δ^-. From the perspective of a selected partition P_s, they indicate to what extent sets from another partition P_x can be considered as merges or splits of sets in P_s.

$$\delta^+(P_s, P_x) = \sum_{S_{s,i}}^{P_s} \sum_{S_{x,j}}^{P_x} \frac{|S_{x,j}| - |S_{s,i} \cap S_{x,j}|}{|S_{x,j}|} \cdot \frac{|S_{s,i} \cap S_{x,j}|}{n}$$

$$\delta^-(P_s, P_x) = \sum_{S_{s,i}}^{P_s} \left(\left(1 - \sum_{S_{x,j}}^{P_x} \left(\frac{|S_{x,j} \cap S_{s,i}|}{|S_{s,i}|} \right)^2 \right) \cdot \frac{|S_{s,i}|}{n} \right)$$

Both measures are asymmetric and produce values in the range of $[0, 1)$, with higher values indicating higher degrees of merges or splits, respectively. The merge measure δ^+ is based on the weighted sum of overlap fraction for pairs of sets with non-empty intersection $|S_{s,i} \cap S_{x,j}| > 0$. The split measure δ^- is based on the weighted sum of (a slightly modified) Gini Coefficient [Gin12]. It considers, for the splits of a set $S_{s,i}$, not just into how many splits exist for it in P_x, but rather how the splits are distributed.

The two measures are visualized as a butterfly bar chart, with the merge measure δ^+ pointing up and the split measure δ^- pointing down. This also allows reading the sum of the two, giving a sense of overall dissimilarity with the selected partition. Figure 4 illustrates the interpretation of δ^+ and δ^- for an artificial dataset. The butterfly bar chart clearly hints at sets from the selected partition P_3 merging in P_1, P_2, and P_6, and splitting in P_4, P_5, and P_6. For merge events, δ^+ is higher when larger sets merge in P_1 ($\blacksquare \cup \blacksquare$), compared to when a large set merges with a smaller set in P_2 ($\blacksquare \cup \blacksquare$). For split events, the middle set ($\blacksquare$) splitting into three subsets in

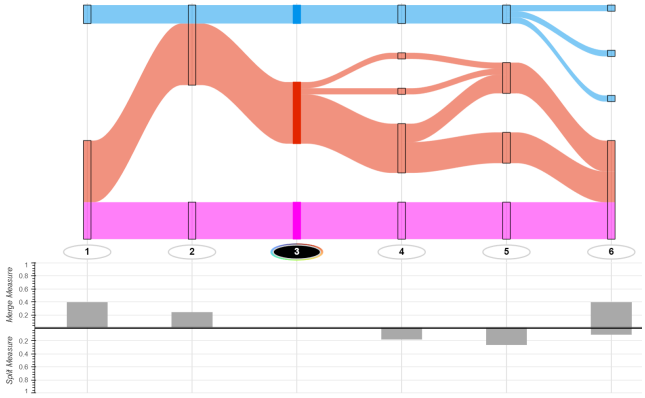

Figure 4: *Butterfly bars indicate the merge measure δ^+ (upwards) and split measure δ^- (downwards) in relation to the selected partition P_3 of an artificial example.*

P_4 has a slightly lower value for δ^- than splitting into two subsets in P_5, reflecting the consideration of the cardinality of the split sets.

Comparisons can also be made on a filtered version of the selected partition to focus on specific subsets. This filtering, triggered by interaction in the alluvial diagram, temporarily removes the filtered-out data items from all partitions and computes the measures on the filtered population. Finally, the $\mathbf{1} \times \boldsymbol{N}$ Comparison View shows nothing if the overview perspective of the alluvial diagram is active.

3.4. $\boldsymbol{N} \times \boldsymbol{N}$ Partition Comparison View

To go beyond the perspective of a selected partition, but to provide a holistic view of partition similarities, it is necessary to compare all partitions simultaneously. The $\boldsymbol{N} \times \boldsymbol{N}$ Partition Comparison View shown in Figure 1(C, top) was designed to not only identify clusters of similar partitions, but also to reveal patterns in the change of partition similarity across the sequence. To achieve this, we compute the pairwise dissimilarity distance between each pair of partitions using the *adjusted rand index* (ARI) [HA85].

$$d(P_x, P_y) = 1 - \frac{ARI(P_x, P_y) + 1}{2}$$

This distance between the partitions lies in the range of $[0, 1)$. Next, similar to our solution of the coloring of sets, using the distance matrix for all pairs of partitions, we run *multi-dimensional scaling* (MDS) [Tor52] to compute a two-dimensional embedding. In this embedding, similar partitions are placed nearby, and dissimilar partitions are placed farther apart. This embedding is visualized as a scatterplot with each point (partition P_i) labeled by l_i. Subsequent partitions are connected by an edge for the perception of sequence. For instance, in Figure 1(C; top), partitions 3, 4, and 5 are further apart than the partitions 5, 6, and 7, which form a dense cluster.

3.5. $\boldsymbol{N} \times \boldsymbol{N}$ Set Comparison View

Finally, in the $\boldsymbol{N} \times \boldsymbol{N}$ Set Comparison View shown in Figure 1(C, bottom), the focus is on a holistic comparison of sets. This view builds on an MDS projection of sets for color assignment in the alluvial diagram and shows further details. Similar to the $\boldsymbol{N} \times \boldsymbol{N}$ Partition Comparison View, it visualizes this 2D projection with sets represented as points in a scatterplot. However, labels are not available and sequential paths cannot be drawn due to changing sets. Instead, a circular sector centered around each point visually encodes the partition index i through its angle. The sector and point are colored corresponding to the set, with the sector's opacity denoting selection status (selected: ▶; not selected: ▶). Additionally, on filtering due to set selection, the glyphs corresponding to sets with none of its items in the filtered population are temporarily hidden.

4. Application Examples

Next, we investigate how the suggested approach could be used to find insights in real-world data. We selected three examples to cover different domains and data characteristics. They include applications in supervised and unsupervised machine learning, as well as social network analysis. Partition sequences are derived from the variation of a parameter or stem from temporal changes.

4.1. Projection Ambiguity of High-Dimensional Clusters

Dimensionality reduction is an established approach to simplify high-dimensional data with $N \gg 3$ dimensions to a more assessable dimensionality $n < N$. Due to the necessarily lossy reduction, it is unclear how the procedure preserves the original data. Comparing the high-dimensional data to the reduced n-dimensional data through clustering, a dimension-independent technique, allows us to assess the impact of dimensionality reduction. We analyze a partition sequence where each partition corresponds to a clustering result, and the dimensionality n of the projected data serves as the label of the partition. As an example, we consider the MNIST dataset [LeC98] with $N = 784$ dimensions and apply *principal component analysis* (PCA) to reduce it to different n-dimensional spaces ($n = 2, 4, 14, 24, \ldots, 784$). The MNIST dataset consists of images representing 10 handwritten digits (classes) and hence is expected to have inherent clusters. Rather than clustering into 10 clusters, we utilize k-means clustering with $k = 25$ clusters. This approach aims to form smaller clusters, trying to avoid a large cluster containing two classes of digits. To compare how the class labels of the dataset are distributed in the different clustering partitions, we add the ground-truth labeling as a first partition.

In the alluvial diagram (Figure 5 top), we observe that k-means clustering is significantly affected by outliers, especially for $n > 400$. Also, due to the massive information loss at low dimensions, we observe substantial changes in neighboring partitions, particularly when shifting from 4 to 2 dimensions. Additionally, we notice that the red stream representing one class does not merge much with other classes across all dimensions, suggesting it is the most separable class when using PCA projections. Conversely, the uppermost two classes are assigned similar colors, although they are disjoint sets. The similarity in color indicates that they are often grouped into the same cluster and may have a higher similarity. The merge measure regarding the original classes in the $\mathbf{1} \times \boldsymbol{N}$ comparison view (Figure 5 bottom) is decreasing almost monotonically from 784 to 24 dimensions, then increasing again—class separability might be highest in the 24-dimensional PCA projection.

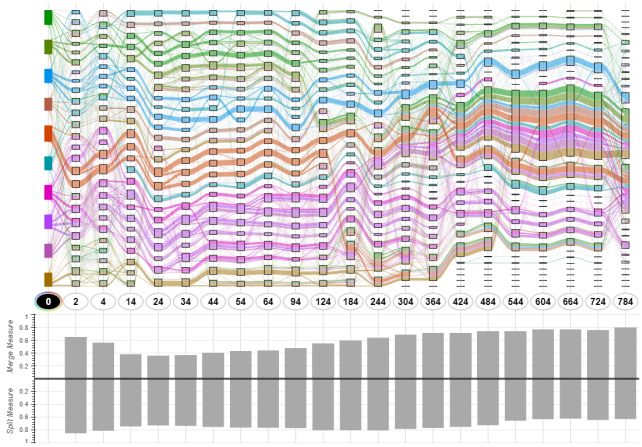

Figure 5: *MNIST data clustered at various PCA-projected dimensions, with the ground truth as the first partition being selected.*

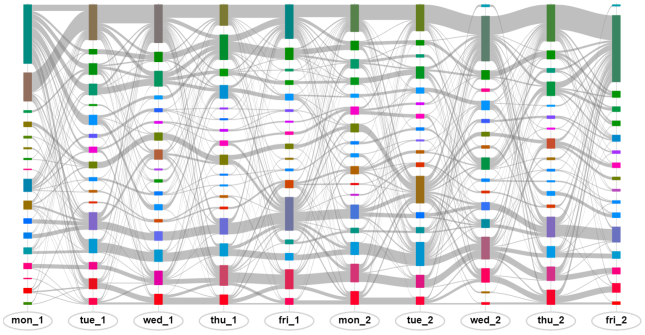

Figure 6: *Contact clusters in the French Health Observatory data shown in the overview perspective.*

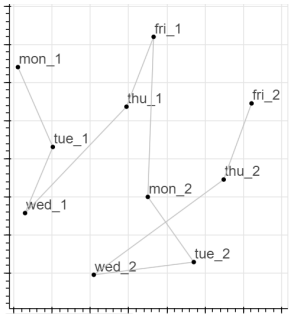

Figure 7: *Similarities of workday contacts in the French Health Observatory data shown in the* $\boldsymbol{N} \times \boldsymbol{N}$ *Partition Comparison View.*

4.2. Contact Cliques Change over Time

As alluvial diagrams were initially introduced to visualize how social network cliques change over time, we demonstrate our tool on contact cliques at a workplace [GB18]. At the French Health Observatory in 2015, RFID chips registered over two weeks whether two persons had a face-to-face contact longer than 20 seconds. To focus on prevalent patterns, we aggregated the resulting contact network over daily time windows via spectral clustering on the adjacency matrix with 16 clusters.

In the alluvial diagram (Figure 6), some large clusters stand out, particularly on the first day, the first Friday, and the last day (*mon*_1, fri_1, and fri_2, respectively). These clusters could indicate co-located events. Additionally, we observed that data items belonging to the red-colored clusters rarely come into close contact with those in the green clusters, except on the three days mentioned. This observation was confirmed through set-based filtering. Overall, the vertical ordering and coloring of the sets helped indicate communities among the participants. Furthermore, in the $\boldsymbol{N} \times \boldsymbol{N}$ Partition Comparison View (Figure 7), a clear margin can be drawn between *mon*_1-fri_1 and *mon*_2-fri_2, indicating that the contact patterns of the first week were significantly different from those in the second week. Further, the Wednesdays, Thursdays, and Fridays each are arranged close to their counterparts; hence, there are recurring similarities between these days in the two weeks.

4.3. Multi-label Classifier across Training Epochs

As our final example, we examine the changes in prediction during training of a multi-label classifier. Agarwal and Beck [AB20] also analyzed this dataset using a stream-based visualization. Their method is tailored to labeled sets (i.e., which allows tracing of specific sets over time), while our approach is limited to non-labeled sets. Nevertheless, we aim to determine if additional insights can be obtained from the same dataset through our approach.

Similar to the original analysis [AB20], we focus on the last training epochs, 22 to 30, where the classifier's accuracy remained relatively stable, fluctuating between 68% and 70%. From the alluvial diagram in Figure 8, we observed that, for the larger selected sets in Epoch 27, there are substantial changes across the epochs in both directions, consistent with the original findings [AB20]. Additionally, the $\boldsymbol{N} \times \boldsymbol{N}$ Set Comparison View (Figure 9 left) with a few sets selected from the last epoch, shows interesting patterns ( ) that indicate that many sets toggle back and forth between memberships. These sets represent data items for which the classifier frequently changes the assigned labels, indicating areas that may warrant further investigation. Lastly, to verify if the toggling behavior in sets also applies to partitions, we examine the $\boldsymbol{N} \times \boldsymbol{N}$ Partition Comparison View (Figure 9 right), which demonstrates that the partitions do not exhibit this behavior.

5. Discussion and Future Work

The three application examples illustrate that our approach helps uncover valuable insights about the data partition sequence, but also hint at limitations. In this section, we critically reflect how the solution meets its intended purpose and identify areas for improvement.

Alluvial Diagram: The alluvial diagram itself was central to all three application examples. Enhancements such as color allocation of sets and partition/set selection helped detect insights, particularly in non-neighboring partitions. The vertical ordering of sets, edges, and sub-edges facilitated tracking streams across the diagram. However, the MDS algorithm used for color allocation does

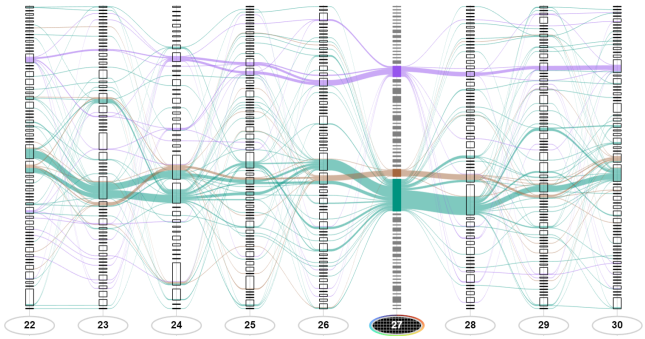

Figure 8: *Multi-label classifier data depicted as alluvial diagram with several larger sets in Epoch 27 selected.*

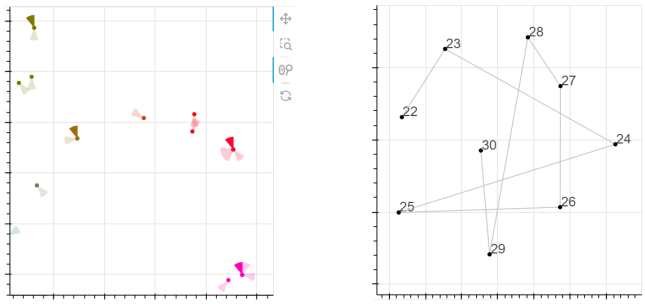

Figure 9: *Multi-label classifier data shown in the $N \times N$ Set Comparison View (left) with selected sets in the last epoch and $N \times N$ Partition Comparison View (right).*

not account for set cardinality, leading to smaller sets having equal weight as larger ones as seen in our first application example (Figure 5). Additionally, this coloring approach can cause dissimilar sets to appear similar due to information loss in the projection. While the former issue is realistic to address in an enhanced algorithm, the latter one is an inherent problem of projection-based approaches. Moreover, in some cases, it might be desirable to select sets across different partitions for comparison, but this would require a complex interaction mechanism and visual encoding to consider multiple intersections, unions, or a combination of both.

Comparison Views: Each of the augmenting comparison views helped reveal, in different application examples, patterns that might have been overlooked with the alluvial diagram alone. Each view offers a distinct perspective on the data, but their usefulness depends on the data. Enhancements could include improved interactive linking, for instance, enabling selections from the scatterplots, which would be particularly beneficial for selecting sets exhibiting interesting patterns in the $N \times N$ Set Comparison View.

Applicability: Our approach visualizes any data expressed as a sequence of partitions. However, there are related dataset and special dataset characteristics that would call for modifications. For instance, to handle named sets, we could vertically align sets with the same name. Further adjustments are necessary to address scenarios involving noise points in clustering results or points lacking set membership in some partitions. This entails changes in visual encodings and similarity calculations between sets or partitions. Additionally, adjustments are needed for fuzzy partitions or partitions that change continuously, rather than in discrete steps. In addition, there is potential for improvement in item-level tasks, for example, implementing highlighting or filtering based on the selection of one or more items.

Visual Scalability: While the approach is decoupled from the number of data items, its *visual scalability* [RPA*24] can be affected by the number of sets (across all partitions) and the number of partitions in the sequence. The application examples discussed involve up to a few hundred sets (539 in the third example). However, as the number of sets increases to the order of thousands, views that visualize each set individually, particularly the $N \times N$ Set Comparison View, will become ineffective. Additionally, frequent and numerous changes in data partitions can lead to numerous edge crossings in the alluvial diagram, making it harder to distinguish the streams. In such cases, the $N \times N$ Set Comparison View still helps identify some patterns. Moreover, increasing the number of partitions to more than 30 would necessitate substantial modifications to the alluvial diagram's visual encoding and interactivity. However, since the $1 \times N$ Comparison View and the $N \times N$ Partition Comparison View aggregate the data, they remain more usable compared to the other two views in such scenarios.

6. Conclusion

With the idea of going beyond tracing partitions' changes from one step to the next, we have extended alluvial diagrams through improvements in the diagram itself as well as linked views. The former includes a similarity-based color encoding of sets and interactive selections, already incrementally improving the tracing of changes ($1 \times \ldots \times 1$) and the comparison of non-neighboring sets ($1 \times N$). The latter provide specific perspectives on visual comparisons that contrast a selected partition to all others ($1 \times N$) or all partitions and their contained sets to each other ($N \times N$). Our application examples not only demonstrate a broad applicability of the technique, but also give an impression of how the different extensions work together to provide a more comprehensive analysis of data partition sequences.

7. Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project numbers 517412225 and 460865652.

References

[AB20] AGARWAL S., BECK F.: Set Streams: Visual exploration of dynamic overlapping sets. *Computer Graphics Forum 39*, 3 (2020), 383–391. doi:10.1111/cgf.13988. 2, 6

[BMBW15] BURCH M., MUNZ T., BECK F., WEISKOPF D.: Visualizing work processes in software engineering with Developer Rivers. In *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)* (2015). doi:10.1109/vissoft.2015.7332421. 2, 3

[BNRB21] BOLTE F., NOURANI M., RAGAN E. D., BRUCKNER S.: SplitStreams: a visual metaphor for evolving hierarchies. *IEEE Transactions on Visualization and Computer Graphics 27*, 8 (2021), 3571–3584. doi:10.1109/tvcg.2020.2973564. 2

[Bok18] BOKEH DEVELOPMENT TEAM: *Bokeh: Python library for interactive visualization*, 2018. URL: https://bokeh.pydata.org/en/latest/. 3

[BZSD21] BARTOLOMEO S. D., ZHANG Y., SHENG F., DUNNE C.: Sequence braiding: Visual overviews of temporal event sequences and attributes. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1353–1363. doi:10.1109/tvcg.2020.3030442. 3

[Chr22] CHRISTOFIDES N.: Worst-case analysis of a new heuristic for the travelling salesman problem. *Operations Research Forum 3*, 1 (2022). doi:10.1007/s43069-021-00101-z. 3

[CLK*17] CHEN W., LU J., KONG D., LIU Z., SHEN Y., CHEN Y., HE J., LIU S., QI Y., WU Y.: GameLifeVis: visual analysis of behavior evolutions in multiplayer online games. *Journal of Visualization 20*, 3 (2017), 651–665. doi:10.1007/s12650-016-0416-0. 3

[CLT*11] CUI W., LIU S., TAN L., SHI C., SONG Y., GAO Z., QU H., TONG X.: TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2412–2421. doi:10.1109/tvcg.2011.239. 2

[CLWW14] CUI W., LIU S., WU Z., WEI H.: How hierarchical topics evolve in large text corpora. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2281–2290. doi:10.1109/tvcg.2014.2346433. 2, 3

[ELAS21] EZELL E., LIM S.-H., ANDERSON D., STEWART R.: Community Fabric: Visualizing communities and structure in dynamic networks. *Information Visualization 21*, 2 (2021), 130–142. doi:10.1177/14738716211056036. 3

[GB18] GÉNOIS M., BARRAT A.: Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science 7*, 1 (2018). doi:10.1140/epjds/s13688-018-0140-1. 6

[Gin12] GINI C.: *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini, 1912. 4

[HA85] HUBERT L., ARABIE P.: Comparing partitions. *Journal of Classification 2*, 1 (1985), 193–218. doi:10.1007/bf01908075. 5

[HHWN02] HAVRE S., HETZLER E., WHITNEY P., NOWELL L.: ThemeRiver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (2002), 9–20. doi:10.1109/2945.981848. 2

[KABB23] KRAUSE C., AGARWAL S., BURCH M., BECK F.: Visually abstracting event sequences as double trees enriched with category-based comparison. *Computer Graphics Forum 42*, 6 (2023). doi:10.1111/cgf.14805. 2

[KBB*23] KESAVAN S. P., BHATIA H., BHATELE A., BRINK S., PEARCE O., GAMBLIN T., BREMER P.-T., MA K.-L.: Scalable comparative visualization of ensembles of call graphs. *IEEE Transactions on Visualization and Computer Graphics 29*, 3 (2023), 1691–1704. doi:10.1109/tvcg.2021.3129414. 3

[KBH06] KOSARA R., BENDIX F., HAUSER H.: Parallel Sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 558–568. doi:10.1109/tvcg.2006.76. 2, 3

[KS98] KENNEDY A. B. W., SANKEY H. R.: The thermal efficiency of steam engines. *Minutes of the Proceedings of the Institution of Civil Engineers 134* (1898), 278–312. Part 4. doi:10.1680/imotp.1898.19100. 2

[LeC98] LECUN Y.: The MNIST database of handwritten digits. URL: https://yann.lecun.com/exdb/mnist/. 5

[LWW*13] LIU S., WU Y., WEI E., LIU M., LIU Y.: StoryFlow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2436–2445. doi:10.1109/tvcg.2013.196. 2

[LYW*16] LIU S., YIN J., WANG X., CUI W., CAO K., PEI J.: Online visual analytics of text streams. *IEEE Transactions on Visualization and Computer Graphics 22*, 11 (2016), 2451–2466. doi:10.1109/tvcg.2015.2509990. 2, 3

[Min69] MINARD C. J.: Carte figurative des pertes successives en hommes de l'armée française dans la campagne de russie 1812–1813. Lithograph, 1869. 2

[OM10] OGAWA M., MA K.-L.: Software Evolution Storylines. In *Proceedings of the 5th International Symposium on Software Visualization* (2010), ACM, pp. 35–42. doi:10.1145/1879211.1879219. 2, 3

[PSB24] PODDAR M., SOHNS J.-T., BECK F.: Not just alluvial. https://github.com/madhavpoddar/not_just_alluvial, 2024. 3

[PW14] PERER A., WANG F.: Frequence: interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (2014), p. 153–162. doi:10.1145/2557500.2557508. 2

[RB10] ROSVALL M., BERGSTROM C. T.: Mapping change in large networks. *PLoS ONE 5*, 1 (2010), e8694. doi:10.1371/journal.pone.0008694. 2, 3

[RPA*24] RICHER G., PISTER A., ABDELAAL M., FEKETE J.-D., SEDLMAIR M., WEISKOPF D.: Scalability in visualization. *IEEE Transactions on Visualization and Computer Graphics 30*, 7 (2024), 3314–3330. doi:10.1109/tvcg.2022.3231230. 7

[RTJ*11] REDA K., TANTIPATHANANANDH C., JOHNSON A., LEIGH J., BERGER-WOLF T.: Visualizing the evolution of community structures in dynamic social networks. *Computer Graphics Forum 30*, 3 (2011), 1061–1070. doi:10.1111/j.1467-8659.2011.01955.x. 2, 3

[TA08] TELEA A., AUBER D.: Code Flows: Visualizing structural evolution of source code. *Computer Graphics Forum 27*, 3 (2008), 831–838. doi:10.1111/j.1467-8659.2008.01214.x. 2, 3

[THM15] TANAHASHI Y., HSUEH C.-H., MA K.-L.: An efficient framework for generating storyline visualizations from streaming data. *IEEE Transactions on Visualization and Computer Graphics 21*, 6 (2015), 730–742. doi:10.1109/tvcg.2015.2392771. 2, 3

[TM12] TANAHASHI Y., MA K.-L.: Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2679–2688. doi:10.1109/tvcg.2012.212. 2

[Tor52] TORGERSON W. S.: Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (1952), 401–419. doi:10.1007/bf02288916. 3, 5

[TRL*19] TANG T., RUBAB S., LAI J., CUI W., YU L., WU Y.: ISToryline: Effective convergence to hand-drawn storylines. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 769–778. doi:10.1109/tvcg.2018.2864899. 2

[VBAW14] VEHLOW C., BECK F., AUWÄRTER P., WEISKOPF D.: Visualizing the evolution of communities in dynamic graphs. *Computer Graphics Forum 34*, 1 (2014), 277–288. doi:10.1111/cgf.12512. 2, 3

[VBW16] VEHLOW C., BECK F., WEISKOPF D.: Visualizing dynamic hierarchies in graph sequences. *IEEE Transactions on Visualization and Computer Graphics 22*, 10 (2016), 2343–2357. doi:10.1109/tvcg.2015.2507595. 2, 3

[WZW*16] WU F., ZHU M., WANG Q., ZHAO X., CHEN W., MACIEJEWSKI R.: Spatial–temporal visualization of city-wide crowd movement. *Journal of Visualization 20*, 2 (2016), 183–194. doi:10.1007/s12650-016-0368-4. 3

[XWW*13] XU P., WU Y., WEI E., PENG T.-Q., LIU S., ZHU J. J. H., QU H.: Visual analysis of topic competition on social media. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2012–2021. doi:10.1109/tvcg.2013.221. 2