SOMFlow: Guided Exploratory Cluster Analysis with Self-Organizing Maps and Analytic Provenance

Dominik Sacha, Matthias Kraus, Jürgen Bernard, Michael Behrisch,
Tobias Schreck, Yuki Asano, and Daniel A. Keim

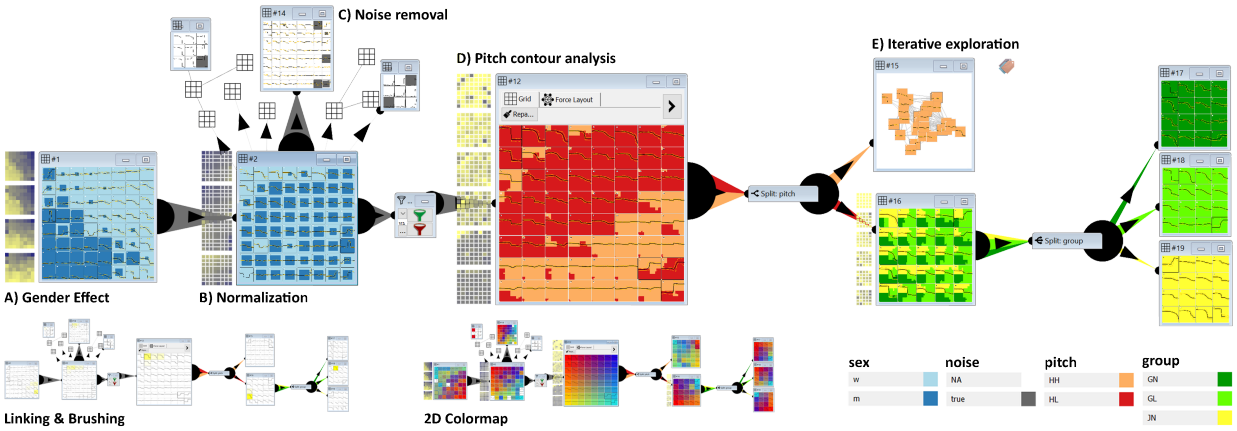

Fig. 1. Overview of a SOMFlow clustering graph that was created during our expert study to analyze speech intonation: First, a gender effect is identified (A) and removed using a domain-specific semitone normalization (B). The analyst created more detailed SOMs for artificial cells and added manual annotations (C) to filter noise caused by measurement errors. The resulting SOM reveals a relation to the pitch meta-attribute (D) and further data partitions allow the analyst to compare pitch contours of different speaker groups.

Abstract— Clustering is a core building block for data analysis, aiming to extract otherwise hidden structures and relations from raw datasets, such as particular groups that can be effectively related, compared, and interpreted. A plethora of visual-interactive cluster analysis techniques has been proposed to date, however, arriving at useful clusterings often requires several rounds of user interactions to fine-tune the data preprocessing and algorithms. We present a multi-stage Visual Analytics (VA) approach for iterative cluster refinement together with an implementation (SOMFlow) that uses Self-Organizing Maps (SOM) to analyze time series data. It supports exploration by offering the analyst a visual platform to analyze intermediate results, adapt the underlying computations, iteratively partition the data, and to reflect previous analytical activities. The history of previous decisions is explicitly visualized within a flow graph, allowing to compare earlier cluster refinements and to explore relations. We further leverage quality and interestingness measures to guide the analyst in the discovery of useful patterns, relations, and data partitions. We conducted two pair analytics experiments together with a subject matter expert in speech intonation research to demonstrate that the approach is effective for interactive data analysis, supporting enhanced understanding of clustering results as well as the interactive process itself.

Index Terms—Visual Analytics, Interaction, Visual Cluster Analysis, Quality Metrics, Guidance, Self-Organizing Maps, Time Series.

1 Introduction

Clustering can be used to analyze large unknown collections of time series data, such as stock market prices, temperature changes, movement features, or spoken utterances, to form subsets of similar data items and to reveal otherwise hidden patterns (e.g., cluster properties and relations). However, analysis problems are often ill-defined or imprecise (neither knowing where or what to seek), interesting patterns (e.g., relations to further metadata) are hidden within particular subsets, and it remains a problem to identify relations among a series of obtained clustering results. Furthermore, the underlying computations need to be adapted to reveal the desired structures for the analysis task at hand.

Hence, this large problem space specifies a need for interactive data exploration in different "directions". Visual Analytics (VA) aims to provide the analyst with a visual platform to explore automatically obtained results to form and refine hypothesis and to interact with the underlying computations if necessary [28, 44]. Tightly intertwined solutions (computations, visualizations, interactions) are needed to cope with nowadays real-world analysis problems [42,43,45] and users which are typically experts in their domain, but novices when it comes to VA, require specific guidance during exploration [17].

To cope with these challenges, we propose an interactive partition-based clustering approach that allows the analyst to drill down into subsets of interest (top down, divide & conquer) based on different division strategies. This approach emerged from our ongoing collaborations (started 3 years ago) with linguistic researchers from the domain of prosodic research (i.e., speech intonation) analyzing time series data of recorded speaker utterances [4, 41]. Our initial VA system used the Self-Organizing Map (SOM) algorithm to create data overviews and it

- *Dominik Sacha, Matthias Kraus, Michael Behrisch, and Daniel A. Keim are with the University of Konstanz, Germany. E-mail: forename.lastname@uni-konstanz.de*
- *Jürgen Bernard is with TU Darmstadt, Germany. E-mail: juergen.bernard@gris.tu-darmstadt.de*
- *Tobias Schreck is with Graz University of Technology E-mail: tobias.schreck@cgv.tugraz.at*
- *Yuki Asano is with the University of Tübingen E-mail: yuki.asano@es.uni-tuebingen.de*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

iteratively enabled the analyst to select data subsets of interest. However, we observed that our users were sometimes overwhelmed by the number of obtained SOMs requiring a visual overview of the analysis process as well as user guidance to support costly and time-consuming analysis tasks (e.g., relation seeking or data annotation).

Inspired by existing hybrid visualization systems (e.g., [27, 51]) we developed the idea to embed interactive SOM visualizations into a graph structure as an analysis overview representing the clustering and interaction flow that further supports higher-level human analytic activities (e.g., organizing and memorizing what has been done, comparison-, or verification tasks). Our resulting SOMFlow system supports four abstract exploration tasks with a rich set of visualization and interaction techniques to (1) analyze and assess the quality of the obtained clusterings, to (2) adapt the computations, and to (3) create further data partitions while (4) keeping the overview. We further leverage quality and interestingness measures to guide the analyst. Hence, we contribute with a general clustering approach and an implemented SOMFlow system (focusing on the SOM algorithm and time series as primary data) that tightly integrates interactive visualization, machine learning (ML), and quality measures, embedded into an analytical reasoning space representing the analysis process.

Next, we provide background information and discuss related work before we describe our approach in detail along four abstract exploration tasks (Section 4). In Section 5, we explain different ways to guide the user during the analysis and describe our SOMFlow system in Section 6. We report on two pair analytics experiments that evaluate our SOMFlow system in a real-world setting (Section 7). Finally, we discuss remaining issues and enumerate promising future research directions (Section 8), before we come up with a conclusion (Section 9).

2 Background on Self-Organizing Maps

We chose the SOM algorithm (also known as Kohonen Maps [31]) as a fundamental approach to automatically generate data overviews. The algorithm introduced by Kohonen has been widely applied to clustering problems and data exploration [29]. It is based on a neural network that can be represented as a grid of cells (neurons, or tiles). Each cell contains an artificial vector (e.g., time series) with the dimensionality of the input data. During the training phase, the vectors are subsequently adapted towards the information provided with the input data. In each step the input vector is assigned to the best matching unit (most similar cell), and this cell as well as a subset of spatial neighbors of the grid are modified for better matching [29]. The result is a grid that represents the data based on their prototype vectors. In the final topology, more similar cells are closer and less similar cells will be farther away. As a result, the input data is distributed across the SOM in a similarity-preserving way. In summary, this algorithm provides data reduction (vector quantization with means vectors), dimensionality reduction (two dimensional embedding), and data clustering/classification (assigning data items to cells). Note that in cluster analysis, the data items within a cell do not necessarily form a single cluster as a set of similar cells can be considered as cluster as well.

Depending on the use case at hand, SOM visualizations either directly show the information of mean cell vectors, or use concrete data items for cell representations (e.g., [13, 46, 54, 55]). The SOM algorithm also allows the visualization of the structure of the grid (e.g., neighborhood information) and quality measures. For our collaboration with linguistic domain researchers (e.g., in our previous work [41]), we visualize the artificially created pitch contour of recorded speaker utterances (sound of the pitch over time) as a thick black line in each cell (see e.g., Figure 3). Data items of every cell (in our case real pitch contour vectors) can be shown on demand. In addition, the analyst can inspect relations between clusters and available metadata which can be used to color the cells. A linguistic task is then, e.g., to analyze how often a certain pattern (pitch contour) appears and if it is related to specific speaker properties (e.g., nationality) to identify differences. Many visualization techniques for SOMs exist and have been applied in different domains. Our work relies on such visualization techniques (that mainly focus on single SOMs) embedded into a workflow supporting iterative data partitioning and analysis steps.

3 Related Work

Our work is related to *Visual Interactive Cluster Analysis* in general where we put specific focus on using the SOM algorithm to analyze time series data. The second part describes existing *Hybrid and Provenance Visualization Systems*. Then we focus on *Quality-based Guidance for Visual Exploration* before we emphasize on the novelty of our work.

3.1 Visual Interactive Cluster Analysis

Visual interactive clustering solutions exist for a variety of data types. E.g., the work of Andrienko et al. [2] proposes methods to group movement trajectories, Ruppert et al. [40] describe visual interactive workflows to cluster textual documents, Cao et al. [16] focus on the interactive analysis of multidimensional clusters, and the approach by Nam et al. [34] focuses on high-dimensional data. Some of them also let the user select a specific subset of interest where another subsequent computation of the clustering/classification is applied (e.g., the work by Choo et al. [18]). Furthermore, specific visualization approaches focus on the visual analysis of time series data [1] with different analysis goals (e.g., segmentation, clustering, classification, motif-detection) [37]. With respect to our clustering scenario we focus on the grouping of time series based on their similarities. A further task is then to seek relations between the obtained clusters and metadata attributes (if available) or to apply data annotations (labels) manually.

Many different clustering algorithms exist (e.g., k-means, hierarchical clustering etc. [26]) and have been applied in VA, often in combination with different metrics and a dimensionality reduction step [45] to obtain a two dimensional embedding of the clusters. Therefore, many VA approaches make use of the SOM algorithm [31] that naturally comprises both steps. Vesanto [54, 55] early described several techniques to apply and visualize the obtained results. Existing SOM implementations and toolboxes that offer gird visualization exist (e.g., Java SOM toolbox [1], or som pak [30]) and further interactive VA systems have been developed. E.g., Schreck et al. [46] describe a trajectory clustering system that offers the analyst visual representations to provide interactive feedback to the algorithm. Further works by Bernard et al. focused, e.g., on time series research data [8] and motion patterns [13]. SOM visualizations have also been used to speed up expensive data labeling tasks. E.g., the work by Moehrmann et al. [33] allows users to apply image labels using SOM visualizations. Finally, the predecessor of the presented work [41] proposed an iterative refinement approach of SOM cell selections and computation adaptions to arrive at subset visualizations of interesting speech intonation patterns. However, a lot of results (SOM instances) were produced making it hard for the analyst to compare and reflect the analysis. Furthermore, it only offered a few visualization techniques and did not support the analyst with automatic recommendations based on quality measures.

3.2 Hybrid and Provenance Visualizations

Another area of related works describes hybrid visualization approaches that embed smaller visualization types into another visualization technique encoding a particular structure (e.g., a tree, graph, or network). A famous example is the Node-Trix system by Henry et al. [27] that embeds matrix representations as aggregated nodes within a social network (node-link diagram). A similar approach is adopted in the OntoTrix [6] system. Other techniques embed several different visualizations into a structure representing the data or analytic flow. Gratzl et al. [24] describe the domino system that enables users to apply data subset selections and manipulations using several dependent visualizations. More recently, Stitz et al [48] propose a data workflow-based visualization system for biomedical research. The work by van den Elzen and van Wijk [52] provides a visual exploration method based on small multiples and large singles. The same authors proposed another technique called BaobabView [51] that explicitly uses the structure of a ML algorithm (decision tree) augmented with smaller visualizations that describe, e.g., data distributions and flows.

Further related system can be found under the heading of "data or analytic provenance" that enable the analyst to track the history of

[1] http://www.ifs.tuwien.ac.at/dm/somtoolbox/index.html, accessed 23.03.17

data transformation or interaction steps. Early works of Young and Shneiderman [58] provide a graphical interface for data filter flows to define and analyze boolean queries. Similarly, Elmqvist et al. [22] describe the DataMeadow system that lets the analyst visually construct queries using graphical set representations within a canvas. Another famous example is the VisTrails system by Callahan et al. [15] visualizing scientific workflow evolutions. The GraphTrail system by Dunne et al. [21] is another example that tracks users' interactions and embeds respective visualizations into an "exploration workspace". Other VA systems, such as Jigsaw [32], offer the analyst specific visual components (e.g., the tablet view) to organize, manage, and annotate bookmarked visualizations.

3.3 Quality-based Guidance for Visual Exploration

Another branch of related research concentrates on guiding the user during the analysis and different approaches to build, e.g., mixed-initiative [19] or relevance feedback [7] systems have emerged. However, the basis for such systems are task models, data-, quality- or interestingness measures.

Bertini et. al [14] survey existing approaches that make use of quality measures in high-dimensional data analysis and propose a quality-based analysis framework that includes measures (e.g., cluster, correlation, or outliers) in data and image space. Sips et al. [47] make use of class consistency measures to determine the quality of cluster mappings from nD into low-D (centroid-based and entropies of spatial distributions). Tatu et al. [49] describe further quality measures for different high dimensional visualizations (e.g., scatterplots or parallel coordinates) and the work of Aupetit and Sedlmair compares visual separation measures [5]. General cluster validity measures such as compactness and separation [25], or silhouette coefficient [39] further exist.

We are also aware of SOM-based quality measures to calculate, e.g., quantization errors within cells, or topological errors [31, 36]. It is further possible to visually asses the quality of a SOM result [12] using SOM-grid/network visualizations, such as the u-matrix [50], or s-map (smoothed data histograms) [35]. Further work by Bernard et al. [9, 10] describes approaches to measure the strength of relations between data content and metadata, such as using Simpson's diversity or Shannon entropy measures.

3.4 Novelty and Contributions of our Work

Section 3.1 describes visual interactive works in the VA domain that leverage the SOM algorithm to obtain clusterings but it also reveals that most of the work focused on one single SOM (or clustering) result. Section 3.2 reveals that many hybrid and provenance visualization techniques exist, however, they are rarely integrated with complex ML methods, such as iterative cluster exploration processes. Section 3.3 offers a variety of approaches and measures that can be used to guide the exploration process, however, concrete real-world applications are rarely described. Hence, we contribute with a hybrid approach to support explicit visualization of the clustering interaction process and further leverage quality and interesting measures to guide the analyst during the analytic process demonstrated with a real-world setting.

4 Partition-based Cluster Exploration

A fundamental idea of our approach is the interactive and iterative construction of analysis workflows for the user-centered partitioning of large complex datasets. The SOM algorithm serves as a powerful visual-interactive data partitioning tool that is widely applied and delivers robust results (further described in Section 4.1). The analysis workflows are visually represented within a graph serving as a means to reflect analytical provenance and support workflow navigation. In every analysis step (node of the graph), users are enabled to analyze partitioning results, adapt algorithmic models and parameters, proceed with downstream partitioning routines, or step back to compare previous results. As such, our hybrid interactive graph implements the overview and details paradigm, facilitated with VA support in every step. In addition, the graph at a glance provides provenance information and can be used for the navigation from coarse to fine-grained analysis. The analyst is presented with a visualization of the entire dataset in the beginning of the analysis. Then, the human task is to decide, how the data can be partitioned, or how the computations can be adapted. These decisions result in new SOMs (or new computations) and iteratively enable the human to navigate into subsets of interest.

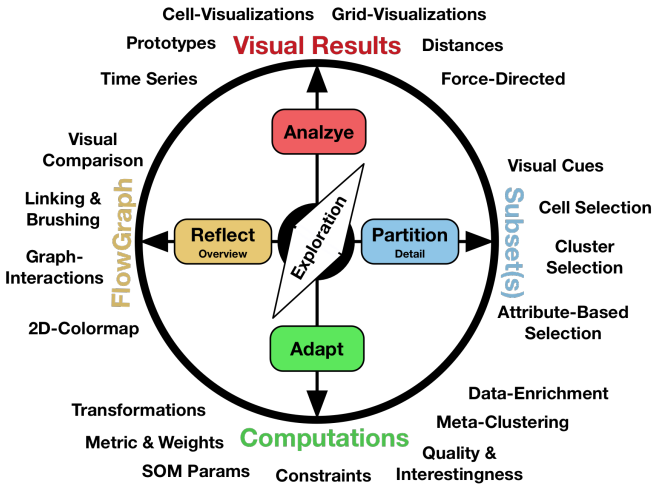

Fig. 2. Our approach supports four abstract exploration tasks: The analyst can **analyze** visual results of the current SOM and **adapt** the computations if necessary. New SOMs are generated with several data **partitioning** approaches. All intermediate cluster results are embedded into a flow graph that enables the analyst to **reflect** the analytic process. For each exploration task, we offer a set of visualization and interactions techniques that are shown outside the circle.

Hence, our approach comprises four abstract exploration tasks enabling analysts to **partition** data, **analyze** data partition results, **adapt** data partition models, and **reflect** the analysis process. We structure them along two orthogonal axes (Figure 2): The vertical axis corresponds to the "current" state of the analysis process that can be visually **analyzed**, as well as **adapted** to improve model and parameter settings. The horizontal axis corresponds to the analysis granularity from coarse (left) to fine grained and detail-rich (right), where new subsets can be created by **partitioning** the data. Exploring the clustering flow graph enables the analyst to **reflect** what has been done in the past. Three of these abstract tasks (analyze, adapt, and partition) directly correspond to building blocks of conceptual VA models (e.g., [28, 42, 43]) while reflect enables higher-level verification activities [42, 44] by comparing the graph elements. Each exploration type is supported by a variety of visualization and interaction techniques shown outside the circle in Figure 2. In the following, we describe these techniques in more detail.

4.1 Analyze Visual Results (Single SOM)

The SOM algorithm is used to enable users to analyze and partition large unknown data collections. Our decision for using the SOM is based on its special characteristics conflating data clustering, vector quantization, dimension reduction, and the ability for cluster visualization [12, 31, 46]. Hence, we make use of existing techniques to support the analysis of the SOM grid visually.

Cells and Time Series: According to the nature of the SOM being a neural network-based clustering algorithm, the output of the algorithm is a (2D) grid, containing a matrix of cells. Each cell represents a portion of the high-dimensional data space, comparable to the cell of a voronoi diagram. The data items mapped to a respective cell are represented by a representant or, like in our case, a means vector that is visualized as a bold black line chart (Figure 3). These vectors are created during an animated training phase allowing users to observe and follow (see Section 7.1) the algorithm and a training history for each prototype vector can be shown as gray background (Figure 3–A). The actual data items that are assigned to the cells are visualized as thinner blue vectors. We also visualize a yellow bandwidth for each cell by drawing the min/max values (Figure 3–B) representing the

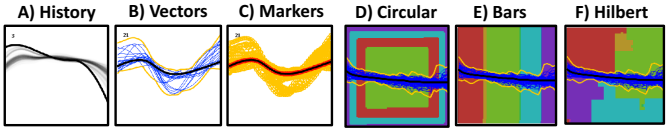

Fig. 3. Cell visualizations: A)–C) depict time series and prototype visualizations. A) illustrates the training history of the prototype, B) renders the prototype vector (black) with the actual time series vectors (blue) and a yellow min/max bandwidth, C) renders the single data points for the prototype (red) and data items (yellow). E)–F) show different pixel-filling techniques to color the cell according to metadata.

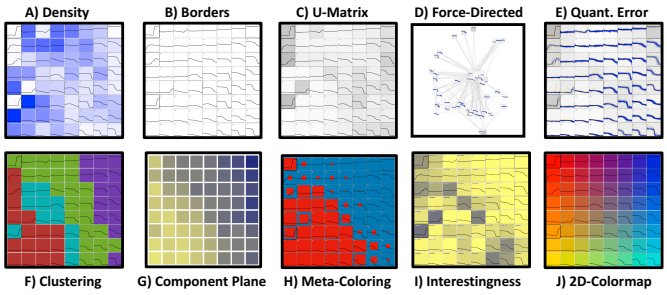

Fig. 4. Grid visualizations: A) Density map, B) distance borders, C) u-matrix, D) cell topology as a force-directed graph, E) quantization error, F) meta-clustering, G) feature component plane, H) meta-coloring, I) cell interestingness, and J) global 2D colormap overlay.

cell uncertainty. Finally, it is possible to show all the data points of each time series step for the prototype (red) and the vectors (yellow, Figure 3–C).

SOM Topology: The grid-based output of the SOM provides the natural structure (topology) of the cells to be visualized (e.g., Figure 4). A series of visualization techniques exist to analyze topological properties in detail. The amount of data items in a cell can be encoded as a density map (Figure 4–A) using a linear color coding (i.e., the more blueish the more data items are contained). Distances between the cells are visualized as black borders within the grid where darker lines indicate larger distances (Figure 4–B). The u-matrix can computed by obtaining the Euclidean distances of all neighbors of a single cell (i.e., it determines how similar or separated a cell is compared to its neighbors). It can be visualized by a gray scale cell coloring where light colors depict most similar neighborships (potential clusters) and dark colors represent more widely separated cells (potential cluster borders or noise, Figure 4–C). In addition, we allow the user to switch to a force-directed graph layout that is using the distances of cell neighbors as spring forces (Figure 4–D). These techniques support the analyst in understanding the topology of the SOM and to identify clusters within the SOM grid. It is also possible to calculate the quantization error (qe) within a cell by comparing all member vectors to its representant/means vector (see also Section 5.1). Visualizing the qe as another gray scale overlay (Figure 4–E) allows the analyst to understand the quality of the quantization (aggregation) within each cell (e.g., dark gray indicates heterogeneous cell content while lighter cells contain data items very similar to the cell vector). We can also apply a meta-clustering (e.g., k-means) to the SOM grid (based on the distances between the cell representants) to automatically obtain and visualize cluster regions on the grid (Figure 4–F).

Component Planes: This technique reveals relations between individual dimensions (components) of the dataset and the SOM result [46]. Given a feature vector representing the aggregated temporal domain, component planes support the comparison of different temporal phases (user-defined set of components/parts) of a time series by coloring each cell according to the average value of the time series (see Figure 9). This allows the analyst to spot specific SOM-regions that have in average high (blue) or low (yellow) values (Figure 4–G).

Visualizing Metadata: Our approach supports the combined analysis of (time series) clusters and additional metadata attributes, such as sensor devices of an experiment, subject measured with a time series recording, distinction between male and female, or the day within the week. If the data contains such additional information its category/value frequency can be visualized. We implemented different meta-coloring techniques, such as circular, bar, or Hilbert pixel filling (Figure 3–D,E,F) to reveal relations between the SOM results and the metadata (e.g, see Figure 4–H). Those techniques produce different visual patterns and are more or less suitable for specific analysis tasks and application domains (see Section 7). The coloring allows the analyst to distinguish homogeneous (single value) and heterogeneous (mixed) cells. With that respect, it is possible to visualize how interesting each cell is using the Simpson's Index (see also Section 5.1) considering a specific metadata attribute. The obtained interestingness value can also be normalized and encoded on the entire SOM grid (see Figure 4–I) [10]. Yellow denotes interesting cells that include a relation to a specific attribute value while dark gray depicts uninteresting cells without relation to the metadata attribute.

4.2 Adapt the Underlying Computations

Users can adapt the underlying computations in every analysis step if the obtained results do not sufficiently meet the analysis requirements. Examples include the wrong preprocessing (e.g., sampling or normalization) of the time series, the distance calculation, or parametrization of the SOM algorithm. Hence, the SOM algorithm may not be able to grasp the desired properties of the data. Our approach offers interactions for each block of the ML pipeline [42, 45] that correspond to the data, the feature space, and the SOM algorithm.

Data: Analysts can assign user-defined labels to each cell. These labels can be used as additional metadata provided by the human (data enrichment) who can, e.g., mark cells (or data items) as "uninteresting" or "interesting". Note, that the computations will automatically consider these labels (e.g., interestingness measure) for providing user recommendations (see Section 5). Iterative data selections are explicitly realized with our data partitioning tasks (Section 4.3).

Transformations: The performance of the analysis depends on the data cleansing and preprocessing strategy. Our tool is able to apply normalization techniques (min/max, logarithmic, square root, etc.) to transform the data values. As the SOM algorithm requires time series vectors of equal lengths, it is also possible to adjust the time series by different strategies, such as simple approaches of adding mean-values (mean–padding) or 0s (zero–padding), or linear interpolation (pair–wise).

Metric and Weightings: The Euclidean distance measure builds the reference metric. We use a weighted variant, allowing the user-based weighting of different temporal intervals of the time series feature vector. The metric can be switched to Manhattan or more expensive computations, such as dynamic time warping (FastDTW) or Earth Movers distance. We also offer an editor to weight different parts of the time series more or less important (Figure 9–C).

Parameter Tuning and Constraints: The SOM algorithm can be parameterized in different ways. Training parameters such as the number of iterations (i), learning radius (r), or learning rate drop (d) can be set in a control panel for each SOM. We animate the training phase by updating the visualization after 1000 steps. Another parameter denotes the form of the SOM (number of cells, rows, and columns). For a default configuration of these parameters we use the "rules of thumb" [31, 55]. Accordingly, we apply a two-step training process: The first step is a rough training ($i = 200000$, $r = 0.3$, $d = 1$) while the second step is a finer training ($i = 1500000$, $r = 0.1$, $d = 1$). Within the training progress, users can fix individual cells of the SOM grid to enforce a specific topology [46]. Adapted configurations can be applied in a new SOM or replace the current SOM. This allows the analyst to compare the different configurations. The parameters can be freely

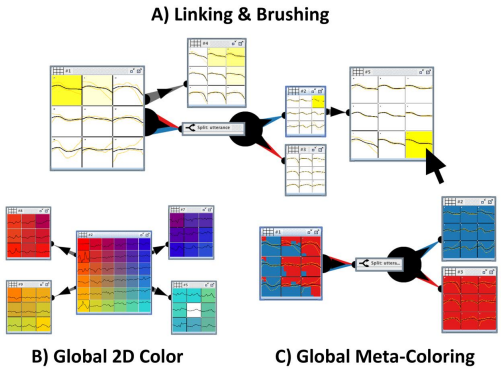

Fig. 5. Graph exploration techniques: A) Linking & brushing across the entire graph on cell hovering, B) A 2D colormap can be created for a selected SOM and shown in the entire graph, C) Meta-coloring for the same attribute for all SOMs.

adjusted based on the analysis task at hand. It is, e.g., possible to create SOMs with different sizes. This allows the user to analyze the same data with different aggregation levels (e.g., step A and B in Figure 8).

4.3 Partitioning Data

Our approach enables the analyst to partition the data into subsets of interest to iteratively arrive at a more fine-grained analysis. We are able to group the partitioning task into three categories.

Cell Selections: The analyst can select cells that either represent an interesting subset of the data that can be investigated in more detail, or depict imprecise cells (high *qe*) that need further refinement in order to arrive at more fine-grained visualizations. Cell selections can be performed based on the primary data and the SOM result quality (Figure 3–A–C, Figure 4–A–E).

Cluster Selections: Similar cells (or cell clusters) that partition the data based on their neighborhoods can be selected to form new subsets. The analyst can either select specific clusters to create new partitions or just split the data based on cluster labels. Clusters can be identified manually supported by the SOM topology visualizations (Figure 4–A–E) or using automatic meta-clusterings (Figure 4–F).

Metadata-Based Selections: On the one hand, metadata can be used to create hypothesis-driven subsets of the data by applying filters or splits based on attribute values. On the other hand, the analyst can seek relations between clusters of the SOM result and metadata. A task for the analyst is to overlay the meta-colorings to reveal specific areas on the grid that are represented by a particular data attribute (see, e.g., Figure 4–H). These attributes (or the respective clusters) can then be used to split the data further. Another reason to partition the data is to select heterogeneous (mixed value) cells (e.g., at the decision border, or outliers cells) to explore more detailed differences within these subgroups. The metadata-based partitioning is supported by the meta-coloring and interestingness overlays (Figure 4–H and I).

4.4 Reflect the Analysis within the Flow Graph

The entire analysis is embedded into a flow graph connecting the SOMs based on their hierarchical relations. It serves as an analytic provenance [57] component that supports higher-level verification activities with a visual comparison of the obtained SOM results.

Flow Graph Elements: Each SOM is a node connected by shared data items (links). The graph is built by the human who is supported by visualizations and quality-based recommendations (see Section 5) to create data partitions. The connections show the number of data items flowing from the parent into the child SOMs. Different elements (direct flow, splitters, and filters) can be created. The arrow size is mapped to the number of data items and metadata can be used to illustrate the data flow.

Interactive Exploration: Our approach offers specific interactions to explore relations within the graph. Hovering a cell will highlight all the cells that contain shared data items. The strength of the highlighting is mapped to the number of common data items (Figure 5–A) by comparing the hovered cell to all other cells of all SOMs. Selecting a specific SOM will highlight the original cells within the parent SOM by adding an orange border around these cells (e.g., Figure 8–B). When users want to analyze distributions of cell contents of an entire SOM with all SOMs of the analysis graph, the 2D colormap technique can be used. 2D colormaps [11] dye the cells of a SOM with similarity-preserving colors, either depending on the input or the 2D output space. We transmit the color-coding to all SOMs in the analysis graph, allowing the lookup of similar cells in different SOMs, as well as the comparison of cluster structures across SOMs. An example for this color-linking strategy is depicted in Figure 5–B. Similarly, it is possible to select a global meta-coloring (Figure 5–C) and finally, we let the analyst switch between a local (per SOM) and global (per graph) min/max-normalization for the data-rendering.

Meta Interactions: In real-world analysis tasks, the flow graph can grow fast and different analysis branches can be created. The graph elements can be re-arranged, resized, maximized, and minimized. It is further possible to navigate within the canvas (zoom & pan) and to annotate graph elements with textual descriptions. This all supports the analysts verification activities, such as knowledge management, adding interpretations, remembering results, and drawing conclusions.

5 Providing Guidance

Our approach offers a rich set of visualization and interaction techniques to support our four abstract exploration tasks. To enhance the usefulness, we integrated a series of guidance techniques to overcome costly investigations of uninteresting data properties.

5.1 Computing Groupings, Quality, and Interestingness

We automatically calculate groupings, quality, and interestingness measures based on data, SOM, and metadata properties.

Automatic Clustering: It is possible to apply a meta-clustering to the SOM-grid that is computed based on the cell prototypes. We implemented different algorithms (k-means, k-medoids, coweb, SOM) that can be chosen by the analyst who may then decide to split the data based on the obtained cluster labels.

Similar Cells: Once a user selects a cell (that is considered as interesting) we can compute if there are neighboring similar cells that can be suggested for extending the selection (*simCells*). We make use of the normalized cell distances (*dist*) to identify the relevant neighbors that have a smaller distance than a similarity threshold $simT = 0.3$.

Cell Quality: We can automatically point the analyst to imprecise cells with a high *qe* as candidates for further refinement. We compute the *qe* according to [31] by calculating the mean Euclidean distance of all cell members compared to the cell prototype vector. *qe* is normalized for each cell over the complete SOM and we introduce a threshold $qeT = 0.1$ to distinguish good from imprecise cells. We also leverage the SOM topology and put neighboring cells with $qe > qeT$ into a common SOM resulting in new child SOMs that show imprecise areas in more detail.

Interestingness: Other measures support the identification of interesting relations between time series clusters and metadata properties. Similar to [10], we calculate an interestingness score for each metadata attribute and SOM cell (i.e., calculate a diversity score of contained attribute-values for each cell) using the Simpson's Index (*simpIdx*). We can make use of this measure to identify interesting metadata attributes with potential relations to the SOM result.

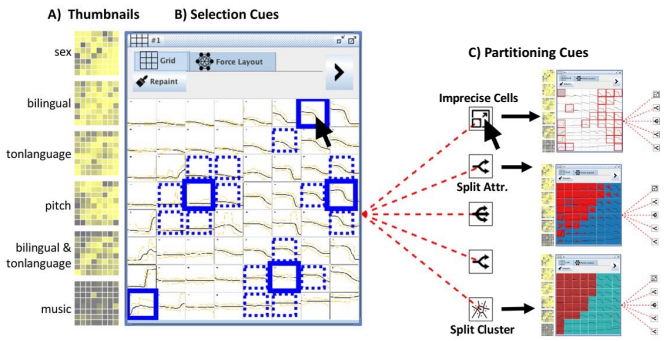

Fig. 6. Visual cues recommend ways to partition the data: A) Ranked attribute interestingness, B) selection extensions, C) partitioning cues.

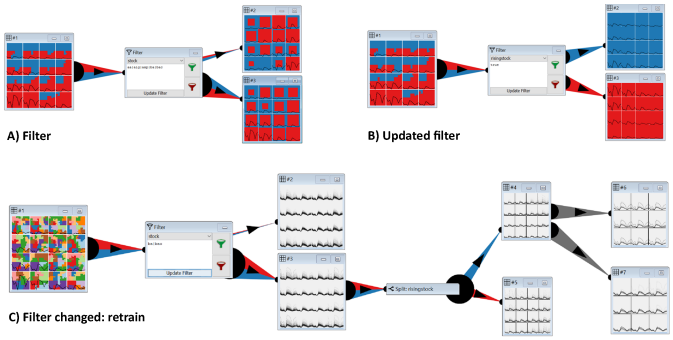

Fig. 7. A) Filter divides data in two parts by some meta-attribute, B) filter is changed with by different constraints, C) filter updates affect all following SOMs in the hierarchy.

5.2 Providing Recommendations with Visual Cues

This section describes how we leverage the described measures to provide the analyst with visual recommendations.

Interestingness Ranking: We append a thumbnails bar at the left hand side of each SOM visualizing ranked attribute interestingness overlays (Figure 6–A). Similar to the approach of Bernard et al. [10], the rank of a meta-attributes is determined by a three-step calculation: 1.) For each SOM cell and metadata attribute we calculate the interestingness (*simpIdx*), 2.) The average interestingness over all cells and attributes is calculated, 3.) For each attribute, we count how many cells are more interesting than the average and we use the obtained value to rank the meta-attributes. If two meta-attributes share the same rank, a second-level ordering is done by considering the average interestingness value of each attribute.

Extending Cell Selections: Once a user selects a cell of interest we obtain relevant neighbors (*simCells*) and visualize a dashed selection border as a visual cue to extend the current selection (Figure 6–B).

Partitioning Cues: We visualize the most significant recommendations for creating further data partitions on the right hand side of the SOM (Figure 6–C). Hovering the partitioning cues will reveal the respective visualization overlay on the SOM grid (Figure 6–right). In case of imprecise cells we show the respective cells with red borders. In case of recommended attribute splits the meta-coloring is shown, and finally, in the cluster split case, we show the cluster coloring. Our recommendation system contains three different types of actions. The first type of recommendation is the SOM refinement by retraining imprecise cells as new SOMs. Therefore it suggests to retrain cells with a high qe ($> qeT$). Neighboring imprecise cells are aggregated and trained in a single SOM. This option is always the first recommendation, if available. Second type of action is the split option. It suggests to partition the data into subsets with equal meta-attribute characteristics based on its interestingness value as high interestingness implies high homogeneity in the SOM cells. Therefore, it might be interesting to analyze each meta-attribute characteristic separately. A maximum number of five meta-attributes with high interestingness average values are suggested ($avgInterestingness > 0.6$) and ranked, similarly to the thumbnail previews. Last but not least, a third type of action recommends to split the data by a meta-clustering. Clicking on any partitioning cue will trigger a data partitioning action.We are well aware that our thresholds appear a bit arbitrary and need to be adapted based on the data and analysis tasks at hand (see Section 8).

6 The SOMFlow System & Use Cases

All the described methods are implemented within our SOMFlow system. In the following, we introduce the remaining details with exemplary use cases.

Implementation: The system is implemented in Java using the Java 2D Graphics API and the Swing library for rendering. The SOM algorithm is also implemented in Java what enabled us to tightly integrate the computations with the visualizations. We further make use of prefuse[2] to generate a force-directed layout and included javaML[3] for basic ML functionalities.

Data Handling: The system can handle any data in JSON format with the only requirement that each data object has to contain a numeric array that can be used as primary (time series) data. As the SOM algorithm requires data vectors of equal length we offer several data processing operations (see Section 4.2). All remaining data attributes are parsed as categorical metadata.

Attribute Manager: The meta-coloring (for single SOMs or the entire graph) can be controlled using the Attribute Manager. This component automatically assigns a default color to each attribute value but also allows to assign custom colors using a color chooser. It is further possible to add new user-defined attributes (adding a name and class labels with colors) that can be used for data annotations.

SOM Interactions: It is possible to interact with the SOM cells: 1.) hovering will trigger the linking & brushing for the respective cell, 2.) left-click will select a cell of interest (and trigger the similar cell recommendations), and 3.) right-click will open a context menu for applying manual data labels. It is further possible to enable global or local rendering options within a control panel next to the canvas. Another controls bar on top of each SOM allows to switch between the grid and force-directed layout, while another controls bar can be revealed on the right hand side of each SOM (e.g., Figure 10–F). This bar offers controls to 1.) annotate the SOM or the graph (adding notes), 2.) create a new SOM for selected cells with a default configuration, 3.) creating a new SOM for selected cells with a custom configuration (a configuration panel to set the data processing, SOM parameters, and metric will be opened), 4.) to define data filters, or 5.) splitters.

Filters: We offer filters for metadata-attributes. Figure 7 shows the application of filters with a stock market analysis example[4]. If a filter is applied on the meta-attribute "stock" (to filter e.g., for specific stocks based on their abbreviations), the SOM is split in two parts: a SOM which contains only data items matching the regular expression in the filter (e.g., stock = $aa|aig|axp|ba|bac$) and a SOM which contains all remaining data items. Links connecting filter and SOMs depict the amount of data flowing in each SOM. Filters can be altered by changing the meta-attribute by which it filters or by changing the regular expression (Figure 7–B). All children are recursively updated and retrained as their data changed (Figure 7–C). This data-update and retraining process is restricted to children which have been created by split, filter, or retrain options. SOMs that have been created based on a selection can not be updated due to the loss of information.

[2]http://prefuse.org/, accessed 24.03.17
[3]http://java-ml.sourceforge.net/, accessed, 24.03.17
[4]http://www.stockhistoricaldata.com/nasdaq, accessed 24.03.17

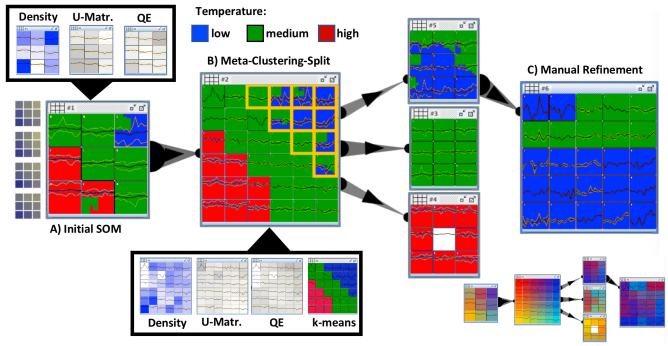

Fig. 8. Yearly temperature changes of the southern hemisphere: A) Initial SOM, B) A bigger SOM is retrained with a meta-clustering, C) cluster labels are corrected manually.

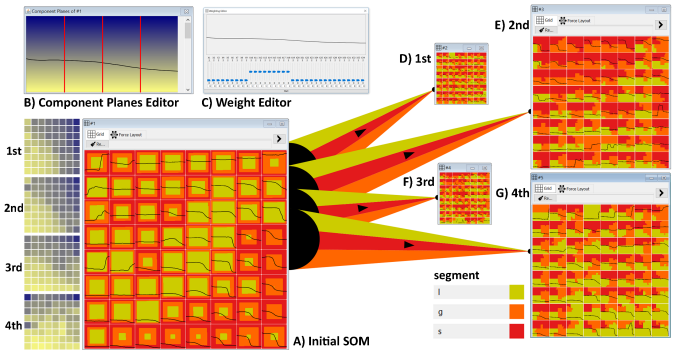

Fig. 9. A) Initial SOM with component plane thumbnails on the left, B) component planes editor allows to generate an arbitrary number of feature components, C) custom weighting for future training can be set, D)–G) component planes 1-4: Each component plane SOM only uses the respective part of the time series for calculation. Visible differences between second (C) and fourth (E) component with regard to distribution of some meta-attribute.

Splitters: We provide data splitters, that divide the data based on attribute values, or meta-cluster labels. An example is shown in Figure 8, where yearly temperature times series (1 value per month) of the southern hemisphere[5] are analyzed (A) and retrained within a bigger SOM (B) to reveal more detailed variations. Then, a k-means meta-clustering was applied to split the yearly temperature progressions (B) and the data has been split to obtain three separate SOMs (high, medium, and low temperatures) that can be corrected manually (C). Note, that the manual annotations are back-propagated within the entire graph.

Component Planes and Weighting. The interestingness thumbnails can be replaced with the component planes on the left side of each SOM (Figure 9–A). The component planes editor (Figure 9–B) and a weight editor (Figure 9–C) can be used to apply configurations. The example in Figure 9–A shows that most prototypes in the bottom left corner of the SOM have relatively low values and high values in the top right corner. We can also see that most prototypes are relatively stable in the first three components (1-3) and change in the end (4). The weight editor (Figure 9–C) can be used to apply custom weights to the SOM training and to generate new SOMs only based on the comparison of a certain part of the time series (Figure 9–D–G).

[5]https://data.giss.nasa.gov/gistemp/, accessed 24.03.17

7 Evaluations

We conducted two pair analytics [3] experiments to analyze empirical linguistic datasets with a subject matter expert (SME) from the domain of speech prosodic research (intonation). The aim was to explore the datasets and to discuss the new system functionality, also compared to our previous version of the system (that focused on one single SOM) [4, 41]. All the used datasets contain a set of recorded utterances for different speakers. We use the utterance pitch-contours (i.e., a curve that tracks the perceived pitch of the sound over time) as primary data for our SOMFlow system and further information about the speaker, utterance, or the experiment as metadata.

Apparatus: One VA expert (VAE1, tool developer) was controlling the system guided by a linguist (SME) who had to interpret the visualizations and point VAE1 to interesting aspects. A second experimenter (VAE2) was observing the study and available for explanations and discussions. We recorded the study, saved important screenshots, and took notes. The system run on a desktop computer using a display with screen resolution of 3840x2160 pixels. The SME was familiar with the basic concepts of our system (SOM, exploratory data analysis) based on our previous collaborations. We also introduced the new SOMFlow functionalities at the beginning of the session.

7.1 Study 1 – Confirmatory Analysis

The first experiment captured data to investigate to what extent a speaker's first language (German or Japanese) influenced the production of intonation when reinforcing an utterance in first and second language. To this end, German and Japanese participants produced the word "Entschudligung" and "sumimsen", both with the meaning "excuse me". The participants had to repeat each utterance three times to attract the waiter's attention within a crowded bar (with the assumption to produce more emotional utterances under increased frustration). Japanese speakers were also learners of German and visa versa. Our SME expected a significant difference between the two speaker groups within the "sumimasen" utterances and we were especially interested in the usefulness of the recommendations that should guide the analyst to answers for this hypothesis.

Dataset: The dataset contains 185 recorded pitch contours (pitch value over time) with metadata about the speaker (e.g., nationality, age, etc.) and the utterance (word, repetition). The time series have been well pre-processed by the SME to make them comparable (smoothed using B-splines [20]).

Tasks and Procedure: We presented the SME with the initial SOM and explained all recommendations. The SME had to comment and assess the quality of each recommendation and the task was to decide which actions are most interesting to pursue in order to derive findings and explanations from the visualizations. We were especially interested in whether the recommendations automatically point the analyst to the predicted differences within the "sumimasen" utterances.

A part of the resulting SOMFlow graph is shown in Figure 10. The first SOM is shown on the very left (#1, without the meta-coloring) and the first system recommendation was to train new SOMs for the cells with high *qe*, however, the SME favored to keep the current aggregation and to look for the other metadata recommendations before. These recommendations pointed the analyst to the attribute "japanology" (indicating if a speaker studied Japanese) where the SME discovered that this attribute is only tracked for one of the speaker groups. To visualize this effect we decided to split the data based on the groups (Figure 10–A) and can reveal that only the German speakers contain "true" values. The SME further reported that the cells with a magenta-color filling look "more Japanese-like". By investigating the recommendations for the obtained subsets we were able to identify further attributes that are only tracked for one of the subgroups or contained coding errors.

The next recommendation was to split the data based on the different utterances (see color overlay in SOM#1). By comparing the obtained SOMs (Figure 10–C) the SME was able to interpret that the

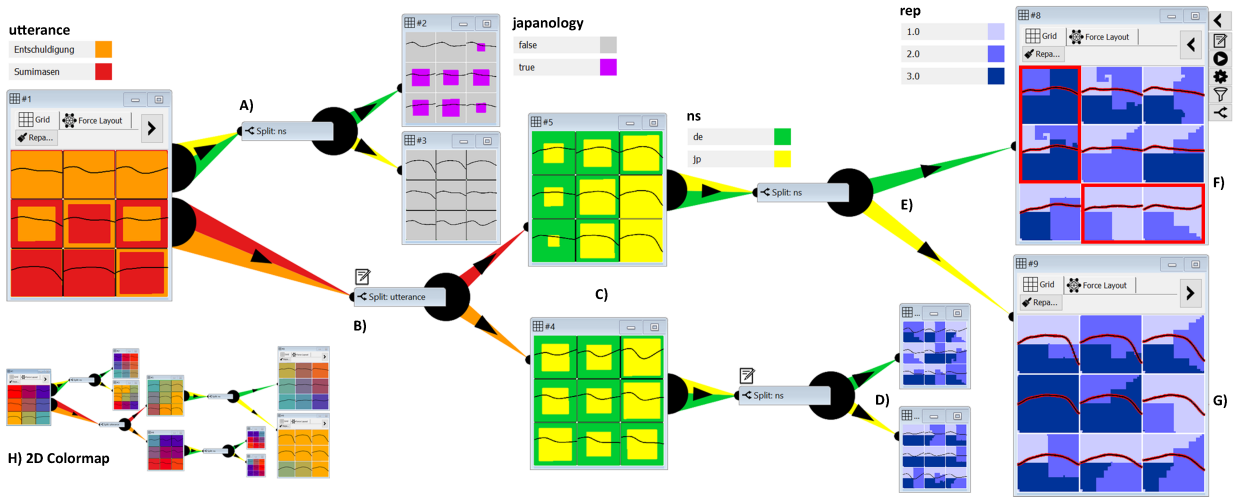

Fig. 10. A part of the SOMFlow graph that has been produced during the first study. An unexpected finding was that some metadata is only present for one of the speaker groups and requires different labels (branch A). The the system automatically pointed the analyst to speaker differences within the "sumimasen" utterances (C – SOM #5) revealing a steeper pitch fall for Japanese native speakers. Further splits and investigations (E) revealed stronger pitch variations (G) for German speakers (because they use pitch to express emotions, in contrast to Japanese native speakers). H) overlays a 2D colormap to compare all SOM cells within the flow graph.

"Entschuldigung" (#4) utterances have more variations than the "sumimasen" (#5) because in German pitch is primarily used to express emotions, while this is not the case in Japanese [53, 56]. To reveal speaker differences, the SME decided to use the nationality color overlay for both SOMs and to split the data further in order to investigate the utterance repetitions (1, 2, 3–blueish colors). As the SME did not find any patterns in the lower branch (Figure 10–D), we focused on the "sumimasen" utterances. The SOM clearly revealed that Japenese (native) speakers have a steeper pitch fall in the end (yellow cells in SOM#5). Splitting the SOM according to the speaker groups (Figure 10–E) also reveals that the contours produced by Germans have a stronger variation (Figure 10–F) than the ones of Japanese speakers (Figure 10–G). The SME concluded that the Germans ignored basic linguistic rules of the Japanese language [53]. Using the blueish color overlay for the repetition attribute also revealed that the Germans produced a raising pitch for the first and second repetition (politeness) and a falling pitch for the third repetition (impoliteness). The SME concluded that the German participants tried to adapt their habits to Japanese. In the end, we used the 2D-colormap to reflect the analysis.

Results: Especially in the beginning of the analysis, we observed that the SME investigated each recommendation in detail. However, during an "analysis branch" the SME formed (novel) specific hypotheses that could be tested by manual color overlays, splits, or cell selections. After these hypotheses have been confirmed or rejected, the SME was able to come back to focus again on the previous recommendations (it was good to have the graph to remember that there was another recommendation). The SME also mentioned that the bar coloring (e.g., see Figure 3–B) is not useful for their domain, because it could communicate a relation between the different parts of the pitch contours and the colors. The study took much longer than expected because most of the system recommendations were interesting and also helped the SME to identify errors in the data (e.g., missing or wrong labels for subsets where the system recommended interesting attributes). We further observed that the SME focused on the attribute-based selections and did not follow up on the SOM or cell properties. We can also confirm that the recommendations automatically pointed the SME to the predicted speaker differences in the "sumimasen" case and that the SME was able to derive further insights about the data. We also observed that the graph grew very fast and the SME liked the ability to reflect the analysis by having an overview with a 2D-colormap and linking & brushing functionality. The note taking functionality was also considered as useful and the animated SOM training with the history overlay helped the SME to get an intuition about its function. The most interesting question of the SME during the study was if it is possible to rate the recommendations with respect to their usefulness depending on the analysis task at hand. In general, the SME was very satisfied with the results and arrived at a useful overall picture of the different groupings within the dataset.

7.2 Study 2 – Exploratory Analysis

We used another bigger and unprocessed dataset to test which functionality of the tool is used and needed to arrive at interesting insights about the data.

Dataset: 7179 utterances of non-sense words (e.g., gubbu, punnu, nunnu, etc.) with High-Low-contour (HL) or High-High-contour (HH) have been imitated by 48 German learners of Japanese (=GL), 24 Japanese (=JN) and 24 German non-learners (=GN). The data contains metadata about the speakers and the pitch contours. They also contain manually annotated labels indicating if the pitch is HL or HH. The HH-condition was considered as a reference for all three participant groups. For the HL-condition, it was expected to find a difference between the groups as the contours in this condition were Japanese-specific. In contrast to the previous dataset, the pitch contours were not normalized nor smoothed. We only applied our linear interpolation processing to obtain vectors of equal length.

Tasks and Procedure: We started with the initial SOM and spotted a gender effect by browsing through the visual recommendation cues. Figure 1–A shows the metadata overlay for the gender information (sex) encoded with blueish colors. We can see that the upper right area is dominated by higher pitch values produced by female speakers and the lower left cells are dominated by male speakers that produce lower pitch values. This effect is also visible within the component planes to the left of SOM#1 where the main difference of these areas appears within the first two components (i.e., first half of the pitch contour). The SME reported that it would be useful to apply a semitone normalization (a domain specific normalization to remove the pitch differences caused by general pitch height differences by female and male speakers). We applied this normalization and obtained a second SOM that clearly visualizes that the gender effect was removed (Figure 1–B), except some cells at the SOM borders that are specific to female speakers. By inspecting the SOM cells (prototypes, bandwidth, and *qe*) the SME spotted artificial contours that could be caused by measurement errors during the experiment. Therefore, we started a noise annotation graph for each selected cell of interest (Figure 1–C) and added a filter to create a subsequent noiseless SOM. The result (Figure 1–D) offers the metadata attribute "pitch" as the most interesting meta-overlay that differentiates red (HH) from orange (HL) contours and the SME reported that the remaining mixed cells

could be checked in more detail to validate these manually annotated labels. However, the SME asked us to split the data based on the pitch label (SOMs #15, #16) and we then focused on the HL data (#16) to further explore the data as it was expected to find differences between the groups in this condition. The SME was interested in the different speaker groups shown as color overlays in SOM#16. Due to the different numbers of data in the three participant groups, we trained separated SOMs for each group. The SME was now able to compare the contours across the SOMs using the linking & brushing functionality to identify speaker differences. In the end, we again zoomed out, activated the 2D-colormap and reflected the analysis steps.

Results: We observed that the first part of the study focused on the data processing, SOM/cell quality, and noise removal while the second part of the study turned over to investigate interesting metadata attributes. During the study the SME identified an uneven distribution within the speakers groups (GN, GL, JN) and the SME reported that it would be useful to see the number of items as further histograms or simple numbers within the attribute manager. Furthermore, the force-directed SOM layout was considered as "a nice feature" but it was not really used by the SME to create subsets. We also observed that the functionality of the system was overwhelming, but the visual cues and the VAEs were able to provide recommendations and explanations. Hence, we conclude that using the system and understanding the concepts/approaches requires training. Finally, the SME emphasized that our approach enables to "freely" explore the data to identify subsets of interest (that include significant effects). These findings could then be verified using conventional statistics.

The two studies demonstrated that our approach was useful to accomplish a variety of analysis tasks. However, we also received useful feedback to improve SOMFlow for the domain of prosodic research.

8 Discussion and Limitations

Our study and ongoing discussions revealed remaining open issues and interesting future work.

We recognized in our user studies that the SME did not fully exploit the functionality of the system (e.g., meta-clustering). Therefore, we aim to improve and fine-tune the recommendations. As a first action, we implemented sliders for our recommendation thresholds (qeT, interestingness rank, $simT$, k-clusters) to steer and test different configurations as an intermediate preparation step to leverage ML techniques to derive good recommendations from explicit user feedback ("guiding the guidance", learning the thresholds). Further improvements can be achieved by considering the current analysis state and previous decisions within a SOMFlow (e.g., by considering already selected/spitted attributes for interestingness calculations). We also envision to experiment with automatically starting computations of subsequent SOMs (or even complete SOMFlow branches) and to investigate how users react to such recommendations. It will also be interesting to revisit, incorporate, and compare other existing automatic approaches to create hierarchical SOMs (e.g., [38]) for our SOMFlow graph (in contrast to our human-in-the-loop approach). Furthermore, we aim to implement "semantic interactions" [23] that automatically adapt the underling computations. E.g., we can automatically adapt the feature weighting based on manual user annotations (using relevance feedback [7]) or enable the analyst to navigate (semantic zoom) through different SOM-grid dimensions.

We focused on categorical metadata and it would be useful to consider numerical attributes as well. On the one hand, we will implement several binning approaches to transfer numeric metadata into meaningful categories. On the other hand, we can offer further metadata color overlays and quality measures for numeric data (e.g., avg/min/max value color encoding). Similarly, the visual design of the SOMs could be further tailored and evaluated for specific data and domain requirements (e.g., removing the bar-coloring for prosodic data or adding other cell visualizations for other data types than time series).

We discussed about focusing in more detail on the analytic provenance aspect of the resulting SOMFlow graph. We could map further data characteristics to the graph (besides link sizes/colors etc.). Similarly, we want to track user interactions for each graph element (e.g., hovers or clicks) that can be mapped to graph properties (e.g., node sizes). Finally, we want to experiment with different automatic layouts (e.g., temporal or SOM similarity based). This will enable us to conduct further studies with the aim to compare and evaluate visual results. Another related aspect are collaborative analysis settings.

We noticed that the resulting graph can be used as a classifier (similar to the decision tree in [51]). It would be interesting to "keep the flow but to change (or enrich) the data" like in common ML scenarios (e.g., cross-validation, training vs. test set). We also noted that our approach "strictly" focuses on a particular dataset that is iteratively partitioned. In contrast, we can experiment with other "flow" paradigms, such as starting from multiple SOMs that merge during the analysis. Another idea would be to freely drag & drop cells to re-assign data. This would, e.g., enable the analyst to create "SOM bins" to organize the data. (e.g., put all good ones into one SOM). However, this would also require to adapt the guidance (automatic recommendations) to these paradigms.

Scalability can be discussed in several ways. Firstly, computation time of the SOM algorithm depends on available recourses of the machine and increases with the size of the data (number of items and vector lengths), the SOM grid (and additional SOM parametrizations), as well as the used metric. Complexity further increases with parallel SOM computations (e.g., after splitting data) and with the quality measures (e.g., attribute interestingness). To avoid long response times, we visualize the iterative process of the SOM training [46], while threading allows to continue the analysis process in parallel to model (re) computations. Threading also allows parallel computation of multiple SOMs. Secondly, the visual and perceptual scalability of the SOM representation depends on screen size and resolution. In case of bigger SOMs (beyond the data sizes of our examples), the cell prototypes or the linking & brushing might not be visible anymore requiring further visualization alternatives to our tile based representations (e.g., aggregates, glyphs, lenses). Thirdly, SOMFlow graphs can become very complex beyond perceptual and cognitive capabilities of the human analyst. Therefore, we can further investigate graph simplification and layout techniques.

Finally, we want to emphasize that our approach is in principle not limited to SOM and could be implemented for other clustering and dimensionality reduction algorithms (or even combine several algorithm types). That would make the approach applicable to a broader range of domains and problems and additionally foster a tighter integration of automatic clustering techniques with interactive visualizations.

9 Conclusion

We proposed a visual interactive clustering approach with an implementation that allows the analyst to iteratively partition the data while keeping the overview. The described SOMFlow system provides a variety of visualization and interaction techniques to support four abstract exploration tasks (analyze, adapt, partition, reflect) and offers additional user guidance. We leverage quality and interestingness measures to provide the analyst with visual recommendation cues and demonstrated their usefulness in a real-world setting. Hence, we were able to derive useful findings about the data and additionally derived interesting future research areas from our observations. As a next step, we will focus on automatic recommendations and fine-tune usability issues with the ultimate goal to offer a powerful and freely available SOMFlow implementation.

Acknowledgments

We gratefully acknowledge the German Research Foundation (DFG) for financial support within the project A03 of SFB/Transregio 161 and within the Research Unit FOR 2111.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[2] G. L. Andrienko, N. V. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 3–10, 2009. doi: 10.1109/VAST.2009.5332584

[3] R. Arias-Hernández, L. T. Kaastra, T. M. Green, and B. D. Fisher. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *44th Hawaii International International Conference on Systems Science (HICSS-44 2011)*, pp. 1–10, 2011. doi: 10.1109/HICSS.2011.339

[4] Y. Asano, M. Gubian, and D. Sacha. Cutting down on manual pitch contour annotation using data modeling. In *Proceedings of the 8th International Conference on Speech Prosody*, 2016. doi: 10.21437/SpeechProsody.2016

[5] M. Aupetit and M. Sedlmair. Sepme: 2002 new visual separation measures. In *IEEE Pacific Visualization Symposium*, pp. 1–8, 2016. doi: 10.1109/PACIFICVIS.2016.7465244

[6] B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev. Ontotrix: a hybrid visualization for populated ontologies. In *Proceedings of the 20th international conference companion on World wide web*, pp. 177–180. ACM, 2011.

[7] M. Behrisch, F. Korkmaz, L. Shao, and T. Schreck. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 43–52, 2014. doi: 10.1109/VAST.2014.7042480

[8] J. Bernard, D. Daberkow, D. Fellner, K. Fischer, O. Koepler, J. Kohlhammer, M. Runnwerth, T. Ruppert, T. Schreck, and I. Sens. Visinfo: a digital library system for time series research data based on exploratory search—a user-centered design approach. *International Journal on Digital Libraries*, 16(1):37–59, 2015. doi: 10.1007/s00799-014-0134-y

[9] J. Bernard, T. Ruppert, M. Scherer, J. Kohlhammer, and T. Schreck. Content-based layouts for exploratory metadata search in scientific research data. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, JCDL '12, pp. 139–148. ACM, New York, NY, USA, 2012. doi: 10.1145/2232817.2232844

[10] J. Bernard, T. Ruppert, M. Scherer, T. Schreck, and J. Kohlhammer. Guided discovery of interesting relationships between time series clusters and metadata properties. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '12, pp. 22:1–22:8. ACM, New York, NY, USA, 2012. doi: 10.1145/2362456.2362485

[11] J. Bernard, M. Steiger, S. Mittelstdt, S. Thum, D. Keim, and J. Kohlhammer. A survey and task-based quality assessment of static 2d colormaps. vol. 9397, pp. 93970M–93970M–16. SPIE Press, 2015. doi: 10.1117/12.2079841

[12] J. Bernard, T. von Landesberger, S. Bremm, and T. Schreck. Multiscale visual quality assessment for cluster analysis with Self-Organizing Maps. In *Proc. SPIE Conference on Visualization and Data Analysis*, pp. 78680N.1 – 78680N.12. SPIE Press, 2011. doi: 10.1117/12.872545

[13] J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2257–2266, 2013. doi: 10.1109/TVCG.2013.178

[14] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2203–2212, 2011. doi: 10.1109/TVCG.2011.229

[15] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Managing the evolution of dataflows with vistrails. In *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDE*, p. 71, 2006.

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

[17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):111–120, 2017. doi: 10.1109/TVCG.2016.2598468

[18] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 27–34, 2010. doi: 10.1109/VAST.2010.5652443

[19] K. A. Cook, N. Cramer, D. Israel, M. Wolverton, J. Bruce, R. Burtner, and A. Endert. Mixed-initiative visual analytics using task-driven recommendations. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 9–16, 2015. doi: 10.1109/VAST.2015.7347625

[20] C. De Boor. *A practical guide to splines*. Springer, New York, 2001.

[21] C. Dunne, N. H. Riche, B. Lee, R. A. Metoyer, and G. G. Robertson. Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *CHI Conference on Human Factors in Computing Systems*, pp. 1663–1672, 2012. doi: 10.1145/2207676.2208293

[22] N. Elmqvist, J. T. Stasko, and P. Tsigas. Datameadow: a visual canvas for analysis of large-scale multivariate data. *Information Visualization*, 7(1):18–33, 2008. doi: 10.1057/palgrave.ivs.9500170

[23] 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

[24] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):2023–2032, 2014. doi: 10.1109/TVCG.2014.2346260

[25] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: Part ii. *SIGMOD Rec.*, 31(3):19–27, Sept. 2002. doi: 10.1145/601858.601862

[26] J. Han, M. Kamber, and J. Pei. Data mining: Concepts and techniques. 2006.

[27] N. Henry, J. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1302–1309, 2007. doi: 10.1109/TVCG.2007.70582

[28] D. A. Keim, J. Kohlhammer, G. P. Ellis, and F. Mansmann. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010.

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

[30] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som pak: The self-organizing map program package. *Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science*, 1996.

[31] T. Kohonen, M. R. Schroeder, and T. S. Huang, eds. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd ed., 2001.

[32] Z. Liu, C. Görg, J. Kihm, H. Lee, J. Choo, H. Park, and J. T. Stasko. Data ingestion and evidence marshalling in jigsaw. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 271–272, 2010.

[33] J. Moehrmann, S. Bernstein, T. Schlegel, G. Werner, and G. Heidemann. Improving the usability of hierarchical representations for interactively labeling large image data sets. In *Human-Computer Interaction. HCI International 2011*, pp. 618–627, 2011. doi: 10.1007/978-3-642-21602-2_67

[34] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. Clustersculptor: A visual analytics tool for high-dimensional data. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 75–82, 2007. doi: 10.1109/VAST.2007.4388999

[35] E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '02, pp. 871–876. Springer-Verlag, London, UK, 2002.

[36] G. Pölzlbauer. Survey and comparison of quality measures for self-organizing maps. In J. Paralič, G. Pölzlbauer, and A. Rauber, eds., *Proceedings of the Fifth Workshop on Data Analysis (WDA'04)*, pp. 67–82. Elfa Academic Press, Sliezsky dom, Vysoké Tatry, Slovakia, 2004.

[37] C. A. Ratanamahatana, J. Lin, D. Gunopulos, E. J. Keogh, M. Vlachos, and G. Das. Mining time series data. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pp. 1049–1077. 2010. doi: 10.1007/978-0-387-09823-4_56

[38] A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Trans. Neural Networks*, 13(6):1331–1341, 2002. doi: 10.1109/TNN.2002.804221

[39] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, Nov. 1987. doi: 10.1016/0377-0427(87)90125-7

[40] T. Ruppert, M. Staab, A. Bannach, H. Lücke-Tieke, J. Bernard, A. Kuijper, and J. Kohlhammer. Visual interactive creation and validation of text clustering workflows to explore document collections. *Electronic Imaging*,

2017(1):46–57, 2017.

[41] D. Sacha, Y. Asano, C. Rohrdantz, F. Hamborg, D. A. Keim, B. Braun, and M. Butt. Self organizing maps for the visual analysis of pitch contours. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015*, pp. 181–189, 2015.

[42] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 2017. doi: 10.1016/j.neucom.2017.01.105

[43] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, D. Weiskopf, S. C. North, and D. A. Keim. Human-Centered Machine Learning Through Interactive Visualization: Review and Open Challenges. *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[44] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1604–1613, 2014.

[45] 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. on Visualization and Computer Graphics*, 23(1):241–250, 2017. doi: 10.1109/TVCG.2016.2598495

[46] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization, Palgrave Macmillan*, 8(1):14–29, 2009. doi: 10.1057/ivs.2008.29

[47] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum*, pp. 831–838, 2009. doi: 10.1111/j.1467-8659.2009.01467.x

[48] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg. AVOCADO: visualization of workflow-derived data provenance for reproducible biomedical research. *Comput. Graph. Forum*, 35(3):481–490, 2016. doi: 10.1111/cgf.12924

[49] 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. on Visualization and Computer Graphics*, 17(5):584–597, 2011. doi: 10.1109/TVCG.2010.242

[50] A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *in Kohonen Maps*, pp. 33–46. Elsevier, 1999.

[51] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pp. 151–160, 2011.

[52] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. *Comput. Graph. Forum*, 32(3):191–200, 2013. doi: 10.1111/cgf.12106

[53] T. J. Vance. *An Introduction to Japanese Phonology*. State University of New York Press, 1987.

[54] J. Vesanto. Som-based data visualization methods. *Intell. Data Anal.*, 3(2):111–126, 1999. doi: 10.1016/S1088-467X(99)00013-X

[55] J. Vesanto. Using SOM in data mining. *Doctoral Dissertations, Licentiate's thesis, Helsinki University of Technology, Espoo, Finland*, 2000.

[56] R. Wiese. *The Phonology of German*. Oxford University Press, Oxford, 2000.

[57] K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, 2015. doi: 10.1109/MCG.2015.50

[58] D. Young and B. Shneiderman. A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation. *JASIS*, 44(6):327–339, 1993. doi: 10.1002/(SICI)1097-4571(199307)44:6<327::AID-ASI3>3.0.CO;2-J