DOI: 10.1111/cgf.14539
Eurographics Conference on Visualization (EuroVis) 2022
R. Borgo, G. E. Marai, and T. Schreck
(Guest Editors)

Volume 41 (2022), Number 3

Exploring Multivariate Event Sequences with an Interactive Similarity Builder

Shaobin Xu[1,2] , Minghui Sun[1,2], Zhengtai Zhang[1,2], Hao Xue[1,2]

[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China

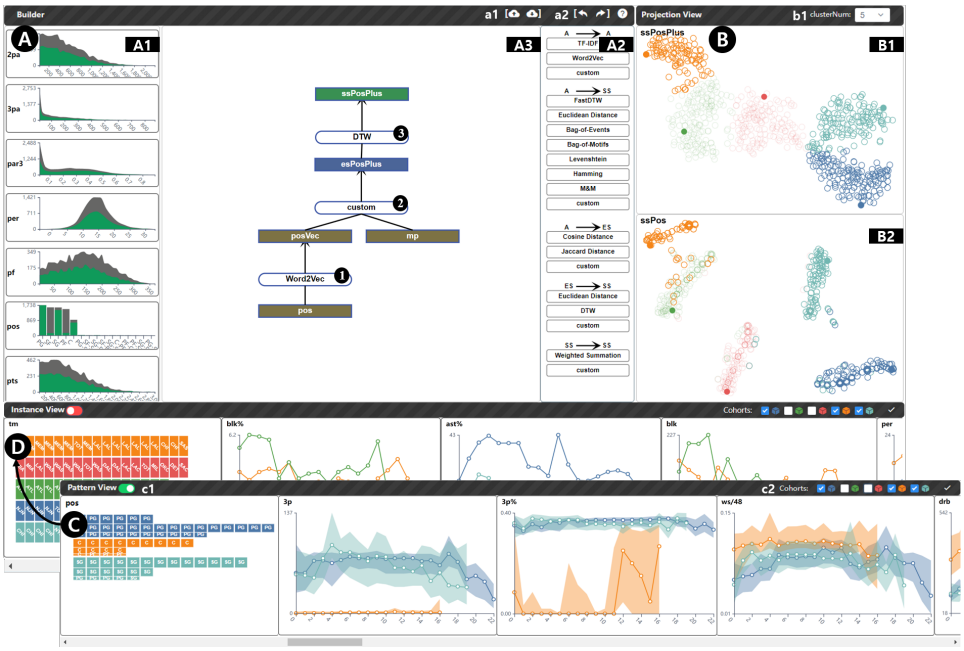

Figure 1: *System Interface. (A) Builder is proposed for visually developing similarity measurements between MVESs. (B) Projection View provides an overview-level validation by capturing and comparing the global structures of the data. (C) Pattern View offers a cohort-level validation by extracting and comparing the temporal patterns of the cohorts. (D) Instance View presents an instance-level validation by displaying and comparing the raw temporal information of the instances.*

Abstract

Similarity-based exploration is an effective method in knowledge discovery. Faced with multivariate event sequence data (MVES), developing a satisfactory similarity measurement for a specific question is challenging because of the heterogeneity introduced by numerous attributes with different data formats, coupled with their associations. Additionally, the absence of effective validation feedback makes judging the goodness of a measurement scheme a time-consuming and error-prone procedure. To free analysts from tedious programming to concentrate on the exploration of MVES data, this paper introduces an interactive similarity builder, where analysts can use visual building blocks for assembling similarity measurements in a drag-and-drop and incremental fashion. Based on the builder, we further propose a visual analytics framework that provides multi-granularity visual validations for measurement schemes and supports a recursive workflow for refining the focus set. We illustrate the power of our prototype through a case study and a user study with real-world datasets. Results suggest that the system improves the efficiency of developing similarity measurements and the usefulness of exploring MVES data.

CCS Concepts

• ***Human-centered computing*** → ***Visualization systems and tools; Interactive systems and tools;***

1. Introduction

Many different domains collect multivariate event sequences (MVESs) from which analysts gain meaningful insights through similarity-based exploration. In addition to two typical attributes, category and timestamp, events in a sequence can also be associated with other multivariate data [GGJ*21]. For instance, medication administration events in the Electronic Medicine Administration Records (EMARs) capture multivariate information (e.g., *timestamp*, *medication*, *dose*, and *care unit*) each time a patient is administered a medication. In real-world applications, the event attributes of such data typically have three common characteristics: (1) large volume, (2) diverse data formats, and (3) complicated association relationships. The success of exploring MVES data depends on the "goodness" of the similarity measurement, which supports the discovery of salient structures in entities. The concept of exploration in this paper refers to the process of obtaining data insights by building and validating similarity measurements.

To date, programming remains the only viable way to create similarity measurements for complex MVES data. However, developing appropriate similarity measurements for diverse analysis questions in this way remains two significant challenges. Firstly, defining and coding similarity measurements can be tedious and require iterative experimentation on different measurement schemes. For MVES data, the existing similarity measurements are either limited to single-attribute information [WS09, WPTMS12, JCG*20] or ignore the association relationships between event attributes [WGW*20, GFL*20]. Given that no universal algorithm can automatically capture these associations, analysts have to tailor similarity measurements according to domain experience. Secondly, although various quantitative metrics for dimension reduction [EMK*19] and clustering [LLX*10,CD18] results are widely used, there is indeed no absolute best metric to evaluate the goodness of similarity measurements independently of the data and exploration questions. Furthermore, the absence of effective validation feedback requires analysts' intuitive judgment to develop what can be considered a satisfiable similarity measurement, which results in each iteration being a time-consuming and error-prone procedure.

Our goal is to help analysts build and validate similarity measurements of MVESs without programming. To address the first challenge, we generalize the unified process of the prevailing MVES similarity measurements and propose an MVES similarity builder, a visual programming tool that allows analysts to use visual data and operation blocks for incrementally composing a similarity measurement of MVESs into an executable directed acyclic graph (DAG). We build the similarity measurements by interactively defining a set of transformation rules in the order of data objects: attribute $\xrightarrow{operation}$ event similarity (Optional) $\xrightarrow{operation}$ sequence similarity. The builder integrates popular algorithms for each type of rule and supports self-defined transformation.

For the second challenge, we design the visualization views to show multi-granularity validation results. The overview-level validation helps users understand the global structure of the sequence entities. The cohort-level validation provides a detailed comprehension and comparison of the cohorts in temporal and multivariate perspectives. The instance-level validation shows and compares the details of individual sequences. To improve the flexibility of MVES data exploration, the visual analytics framework supports a recursive workflow consisting of the following steps: building similarity measurements, validating the performance of the measurements, and selecting a subset of interest to start a new round of exploration.

Specifically, the main contributions of this work are as follows:

- An interactive visual similarity builder where analysts can quickly develop MVES similarity measurements by building data and operation blocks as executable DAGs.
- A visual analytics system that incorporates a set of visualizations to offer multi-granularity validation feedback. The framework supports a recursive workflow to drill down into a particular subset of interest.
- An example usage scenario and a user study that demonstrate the usefulness and efficiency of our prototype using career statistics and hospital treatment datasets.

2. Related Work

Given its broad applicability, event sequence analysis has been extensively studied over the past decades. A comprehensive survey is available in [GGJ*21]. This section provides an overview of the prior work, which is the most relevant to our work, including (1) event sequence exploration, (2) visual programming tool, and (3) multivariate event sequence visualization.

2.1. Event Sequence Exploration

Similarity-centered exploration is an important approach for exploring unforeseen event sequences to discover global structure, potential cohorts, and individuals of interest. Mannila and Ronkainen [MR97] formalized the definition of sequence similarity as the edit distance based on a set of transformation operations (e.g., insert, delete, and move) between events. Similan [WS09, WPTMS12] proposed a temporal categorical similarity measure called M&M, which considered the number of swapping, missing or extra events, and time difference of matched events. Recently, more advanced machine learning techniques have been utilized to facilitate similarity-centric event sequence exploration. CarePre [JCG*20] adopted a distance measure that combines event-to-vector and dynamic time warping (DTW)-based sequence alignment technique [GJG*18] to focus on the population of clinically relevant patients. Guo et al. [GFL*20] introduced a method of calculating the similarity of medical records with event and sequence embeddings. ViSeq [CYP*18] introduced the bag-of-motifs to identify different learner groups, and EventAction [DPSS16] employed the bag-of-events to find similar archived students for action recommendations. In addition, to gain fine-grained insights, the *recursive* nature of data exploration has received increased attention. One striking example for exploring event sequences is MAQUI [LLMB18], where the recursive approach was used to enable interwoven querying and mining to obtain recursive insights.

While various sequence similarity measurements can be applied to MVES data exploration, to the best of our knowledge, none of them can automatically handle all real-world MVES data due to the complexity of the attributes. For such data, tailored similarity measurements are usually developed depending on analysts' preferences and analysis questions. Different from these works, our paper proposes the first visual programming tool for the interactive

development of MVES similarity measurements rather than a specific similarity measurement. Meanwhile, we further present a visual analytics framework that supports a recursive workflow to drill down into the subset of interest or filter irrelevant entries.

2.2. Visual Programming Tool

Aiming at freeing analysts from heavy programming and enabling people with less programming skills to complete development tasks, numerous visual programming tools have been developed in the field of visualization. Nodes on Ropes [WRF*11], MeVisLab [MeV], Inviwo [JSS*19] provided several systems designed to ease the development of tailored visualization applications with visual blocks using DAG. To facilitate the process of creating intra-chart and inter-chart animations, CAST [GLW21] and Data Animator [TLS21] presented two authoring tools that enable users to build chart animations without programming. Built on regular expressions, (s|qu)eries [ZDFD15] proposed an expressive visual query language for building queries on sequences in an approachable way. Users can visually describe high-level patterns of interest by directly manipulating the constraint blocks in 2D canvas and interactively explore the result visualizations. EventPad [CvW17, CMEVW18] allowed users to visually simplify the event sequences into parts relevant to their investigation by specifying rewrite rules. However, the works did not specifically target the development of MVES similarity measurements and their programming tasks are quite different from the task in our paper.

Several visual interfaces [WS09, WPTMS12, WGW*20] supported the user defining sequence similarity for different tasks. However, the usefulness of these systems is limited to supporting only the parameter tuning of a specific metric for a given input data. In practice, faced with complex MVES data and various exploration questions, analysts have to iteratively develop and validate similarity measurements without visual feedback, resulting in a significant programming burden. In our work, our primary motivation is to improve productivity or save time in each iteration consisting of the development and validation of the similarity measurement. Therefore, we use visual building blocks for assembling similarity measurements and validate their results from three granularities: overview, cohort and instance.

2.3. Multivariate Event Sequence Visualization

Recently, given that most real sequence data are multivariate, several MVES visualization techniques have been proposed in various domains to help users obtain comprehensive insights. Timespan [LPK*15] represented all the multi-dimensional and temporal stroke data in a single hybrid view, incorporating factors from multiple visualization components to support exploring the stroke treatment process. EventPad [CvW17, CMEVW18] allowed users to simultaneously explore such data at multivariate and sequential levels by visually defining a set of multivariate rules, which can rewrite MVESs to high-level pattern sequences. To obtain a clear overview of tactical patterns of racquet sports data, Wu et al. [WGW*20] proposed a re-configurable glyph design to simultaneously show multiple attributes of a hit event using six encoding methods. They further designed a glyph-based Sankey diagram [WLG*21] to summarize the ever-changing multivariate tactic progressions. Particularly, multivariate time series data (MVTS) have been widely collected in various fields for studying various phenomena in the real world and they can be seen as special MVES data whose event attributes are all in numeric data format. The techniques for visual analysis of MVTS can be divided into two categories: *Small multiples* and *Large single*. *Small multiples* [GFL*20] juxtaposes individual dimensions to a series of similar charts. *Large single* integrates multiple information in one chart using strategies such as superposition [JME10] and dimensionality reduction [BWS*12, BSH*15].

However, most of the above works were limited to MVES data with attributes of the same data format (i.e., either all-category attributes or all-numeric attributes). Different types of attributes were converted to the same type, which reduced the accuracy of the MVES data. In our work, we extract and then visualize temporal patterns with quantitative uncertainty for each attribute according to their data formats, making them applicable to MVES data in a wide range of domains.

Table 1: *Denotations*

Symbol	Description
$e[a_i]$	a single-attribute event with attribute a_i
$S[a_i]$	a single-attribute event sequence with attribute a_i
$e[A^*]$	a multivariate partial-event with attribute set A^*
$S[A^*]$	a multivariate partial-event sequence with attribute set A^*

3. Background and Requirement Analysis

In this section, we first introduce the data model of MVESs, followed by the process model for similarity calculation. Finally, we present the design requirements for exploring such data.

3.1. Data Model

Like Wu et al. [WGW*20], we adopt a unified data format to normalize MVES datasets for various domains. Each sequence is a specific ordered list of events for each entity, denoted as $S = (e_1, e_2, e_3, \cdots, e_m)$, such as a user's interaction log with an application. We define the set of all event (or sequence) attributes as $A = \{a_1, a_2, \cdots, a_k\}$ and the subset of the attributes as $A^* \subseteq A$. Each multivariate event (i.e., a player's stats for a single season) is described by multiple attribute-value pairs, denoted as $e_i = \left\{a_1 = v_1^i, a_2 = v_2^i, \cdots, a_n = v_n^i\right\}$. Often, users only focus on partial-events, denoted as $e[A^*]$, consisting of partial key-value pairs related to the analysis question at hand. More concretely, the data format of attribute a_j is one type of category, numeric, range, string, vector, set, etc. Some supplementary denotations are shown in Table 1.

3.2. Process model

According to an extensive survey of prior research and our interviews with analysts with over five years of experience, we generalize a unified process model (Fig. 2) for calculating the similarity between MVESs. The process model includes three types of data objects (i.e., attribute, event similarity, and sequence similarity, denoted as the blocks in Fig. 2) and five types of rules (i.e., the links between blocks) based on the type of input and output data objects. Specifically, the similarity measurement computation process is considered as a directed graph that starts from the attribute object

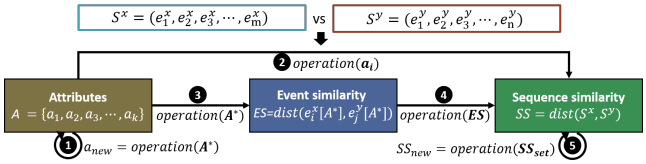

Figure 2: *Process model for MVES similarity calculation.*

and then sequentially executes a series of rules to achieve a specific sequence similarity object. The rule is of the form $\alpha \xrightarrow{operation} \beta$, where α and β are data objects, and *operation* is a certain algorithm or user-defined transformation. That is, the data object α and β are the input and output of the *operation*, respectively.

In this work, these rules are used for building the similarity between two MVESs ($S^x = (e_1^x, e_2^x, e_3^x, \cdots, e_m^x)$ and $S^y = (e_1^y, e_2^y, e_3^y, \cdots, e_n^y)$). The values of each similarity measurement are normalized to be in the range [0,1]. Note that an MVES similarity with a specific purpose is generally related to only a portion of the attributes, denoted as A^*. The five types of rules are as follows:

(1) Attribute → Attribute (A → A)

To obtain additional information, a new event attribute can be derived from the preexisting ones, namely $a_{new} = operation(A^*)$. For example, to obtain accurate trajectories in travel data, a new vector attribute *location* = [*latitude*,*longitude*] can be derived from the attribute *city name* using the *address resolution* operation.

(2) Attribute → Sequence Similarity (A → SS)

Analysts can directly calculate the similarity between sequences based on a single attribute a_i, namely $SS = operation(a_i)$. For example, to identify potential trajectory cohorts, analysts may employ DTW [KR05] to calculate the trajectory similarity between entities according to the new attribute *location*.

(3) Attribute → Event Similarity (A → ES)

In real scenarios, associations between attributes are prevalent and critical for developing sequence similarity. Take, for example, the following EMARs of two patients:

- (Drug A, 200 mg) → (Drug B, 200 mg) → (Drug C, 100 mg)
- (Drug A, 400 mg) → (Drug C, 500 mg) → (Drug D, 300mg)

When we calculate the similarity of these two EMARs, it is necessary to consider both the *medication* attribute and the *dose* attribute, thus avoiding significant loss of information. Furthermore, due to the dependency of these two attributes, it is also unreasonable to split each EMAR sequence into two independent single-attribute sequences in the process of calculating their similarity. Therefore, calculating the similarity of MVESs whose attributes are not independent follows the order of calculating the multivariate event similarity first and then the sequence similarity.

This rule defines how to calculate the similarity of two multivariate events based on two sets of attribute-value pairs, namely $ES = operation(\boldsymbol{A^*})$. The event similarity should consider multiple attributes and the associations between them. For example, the similarity of two medicine administration events requires considering both drug and dose attributes and their dependence.

(4) Event Similarity → Sequence Similarity (ES → SS)

Table 2: *Five types of rules for building similarity measurements*

Rules	Input	Output	Algorithm List
A → A	Attribute	Attribute	TF-IDF, Word2Vec, etc.
A → SS	Attribute	Sequence Similarity	FastDTW, M&M, etc.
A → ES	Attribute	Event Similarity	Cosine Distance, Jaccard Distance, etc.
ES → SS	Event Similarity	Sequence Similarity	DTW, Euclidean Distance, etc.
SS → SS	Sequence Similarity	Sequence Similarity	Weighted Summation, etc.

Based on the multivariate event similarity, analysts can employ some algorithms (e.g., the Euclidean distance and DTW algorithms) to further compute the sequence similarity, namely $SS = operation(\boldsymbol{ES})$, such as the similarity of the EMARs.

(5) Sequence Similarity → Sequence Similarity (SS → SS)

The previous sequence similarities can be combined into a new similarity measurement for a new question ($SS_{new} = operation(\boldsymbol{SS}_{set})$, where SS_{set} is the set of all sequence similarities). For example, the three sequence similarities of 2-point field goals, 3-point field goals and free throws constitute the similarity of NBA players' careers on the offensive end.

3.3. Design requirements

The system is designed for analysts who have experience with different levels of work experience and event sequence mining techniques. Over the course of approximately seven months, we have worked closely with four senior event sequence analysts. The analysts work in a large data company that provides data analysis services for various domains. At the early stage of the collaboration, analysts introduced the MVES data in different domains, the tasks in daily analysis, and the analysis process. Subsequently, we developed an early system prototype and then improved the designs iteratively. In each iteration, we held weekly meetings with the analysts to introduce the visual designs and collect their feedback and comments. Finally, we identified the five design requirements consisting of one similarity measurement development requirement (R1), three validation requirements (R2–R4), and one workflow requirement (R5). They motivate the design adopted in the current version of the visualization system.

R1 Support visual the development of similarity measurements. The traditional development of similarity measurements generally requires tedious programming and multiple iterations. In addition, the definition of what constitutes sequence similarity can change on a question-by-question basis, further exacerbating the programming burden. Analysts hope that the system can provide a more efficient and accessible visual environment that allows users to visually build sequence similarity with drag-and-drop interaction.

R2 Overview and compare the global structures of the data. The visual validation of similarity measurements should start with an overview visualization to identify the global structure and local differences of the data.

R3 Summarize and compare cohorts in temporal and multivariate perspectives. To help analysts to obtain a multivariate comprehension of the cohorts, we need to summarize temporal patterns on different attributes. For each attribute, a generic chart should be designed based on its data format.

R4 Display detailed sequences of the instances. Analysts may examine instances to validate similarity relationships. Thus, an instance view should be provided to display the raw MVES data.

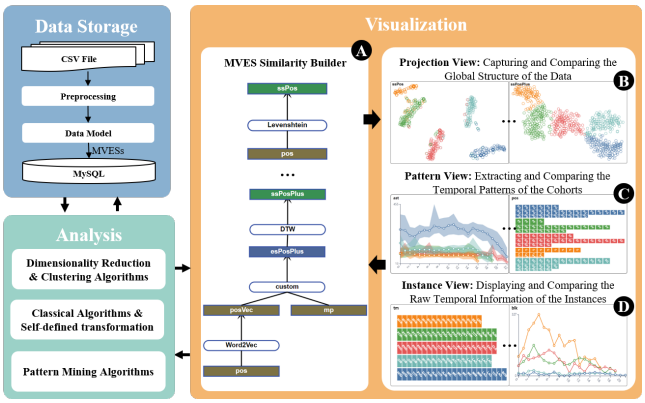

Figure 3: *System overview. It contains three modules: data storage, analysis, and visualization.*

R5 Keep track of recursive exploration history. During recursive exploration, analysts may keep narrowing down the subset of interest for further exploration. To avoid analysts from getting lost in the process, the system should track the steps and reverse the data drill-down steps if necessary.

4. System Overview and Implementation

Fig. 3 provides an overview of our system. The visualization module includes an interactive similarity builder and visual validations at three granularities. When using the system, analysts can first build sequence similarity measurements (R1) in the builder based on the exploration question at hand (A). Then, the results of dimensionality reduction and clustering are automatically calculated and visualized in the projection views (B) for capturing and comparing the global structures of the data for overview-level validation (R2). Next, analysts can understand the cohorts by comparing the temporal patterns of different clusters in the pattern view (C) for cohort-level validation (R3) and focus on cohorts of interest for a new round of detailed exploration (R5). Finally, the raw temporal information of the instances of interest can be displayed in the instance view (D) for instance-level validation (R4).

Our system is an HTML5 web application consisting of a database based on MySQL, a backend based on Django, and a visual analysis interface (Fig. 1) based on Vue.js and D3.js [BOH11]. The database stores user-supplied data (i.e., sequence table in CSV file) imported via Fig. 1 (a1 left). The similarity measurements can be exported as Python files via Fig. 1 (a1 right). The backend implements three categories of algorithms: (1) classical algorithms (e.g., FastDTW [SC07]) and arithmetic functions for the self-defined transformation of the input object to the output object, (2) dimensionality reduction and clustering algorithms for identifying cohorts, and (3) temporal pattern mining algorithms for summarizing cohorts (see the supplemental material).

5. Visual Design

The system comprises four views, namely an interactive similarity builder and three granularities of validation visualizations.

5.1. Similarity Builder

The builder is designed to make it easier to develop similarity measurements for MVES by assembling visual building blocks into an executable DAG (R1). It is an incremental build process where each build action executes a certain rule (see Section 3.2) on existing data blocks to generate a new data block, thereby extending the DAG. The builder consists of the following three components.

Attribute view. Before building the similarity measurements, analysts should initially understand the attributes to discover the ones of interest. On the left of the builder (Fig. 1 (A1)), we employ several basic charts (e.g., bar chart and area chart) to display the distributions on the attributes with different data formats. Once an entity subset of interest is selected, an additional attribute distribution of the subset is appended in each attribute chart, displaying the fraction of the subset. More importantly, the charts are used as interactive panels for analysts to drag and drop corresponding attribute blocks onto the canvas.

Operation bar. To avoid repetitive programming, two types of operation blocks (standard and custom blocks) are integrated into the builder for each rule. First, we review the related algorithms used for sequence similarity, incorporate them into an operation repository as standard operation blocks, and categorize them into five types of rules (mentioned in Section 3.2) based on the type of input and output data objects, as shown in Table 2. Fig. 1 (A2) lists, in rule order, the commonly used operation blocks, which are selected by the user from the repository and represented as rounded blocks. Furthermore, analysts can choose a corresponding custom block and then visually define the transformation from the input to the output of the rule via the formula builder panel (Fig. 4). As with the attribute view, analysts can also drag and drop a corresponding operation onto the canvas.

Canvas. In the 2D canvas (Fig. 1 (A3)), analysts can implement a purpose-specific similarity measurement of MVESs by applying the five types of rules in an executable DAG. As the rules are defined sequentially according to the process (Fig. 2), the DAG extends toward sequence similarity. The attribute, event similarity and sequence similarity blocks are colored brown, blue and green, respectively. The rounded rectangles represent the operation blocks. Each rule, that is, an edge of the executable DAG, is represented by a linked block series (see Fig. 1 (A3 ①②③)). Analysts first select some data blocks, and then choose an operation block to perform on them. Finally, the system automatically calculates a new data block, which can be used as an input to build new rules.

Our system supports direct interaction with the operation blocks to set and modify their details. When the analyst drags and drops a standard operation block onto the canvas, a standard panel will pop up to show different options (see the supplemental material), including the name of the output data object and the necessary parameters of the algorithm. In particular, when existing algorithms cannot meet the analysis demand at hand, the analyst can drag and drop a rule-specific custom block onto the canvas and visually define the transformation formula from the input to the output via the formula builder panel. For instance, Fig. 4 presents the custom panel of rule A $\rightarrow$ ES. The panel integrates common arithmetic functions (b1) and shows the input data objects (b2 and b3) in the toolbar. Operationally, starting with just one space in formula workspace (b4),

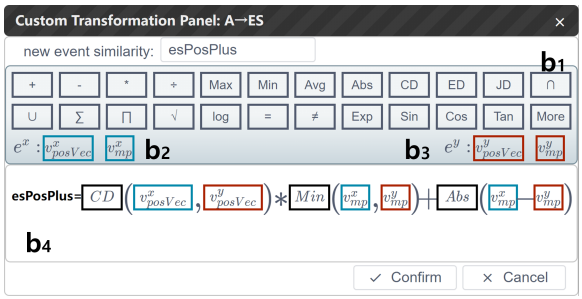

Figure 4: *Custom transformation panel for defining rules.*

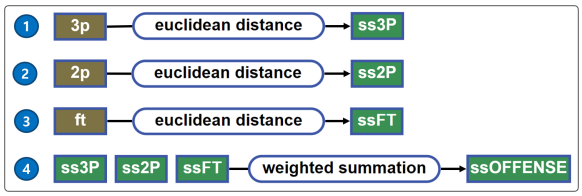

Figure 5: *Design alternative for building similarity measurements.*

the user is required to select a data element or function for each space. When the user selects a function, new spaces are appended according to its input parameters, and its range is enclosed in parentheses. This process is repeated until the needed transformation is finished. The panel can also be integrated as a component in other visual analysis systems and its arithmetic functions can be modified if desired.

Justification. For the iterative design process, we discussed an alternative design of the builder, as shown in Fig. 5. Similar to Eventpad [CvW17, CMEVW18], multiple rules are displayed in a list and applied from top to bottom. A rule can be defined by selecting data object(s) and an operation object via a drop-down menu. However, the design may impose a heavy cognitive load on the analysts as they have to mentally reconstruct the similarity calculation process based on the list of rules. Compared to the rule list design, the DAG is a simple yet powerful tool for creating a new rule while keeping a mental map of the process. Furthermore, the directed graph helps analysts make associations between data and operation objects through drag-and-drop interaction, which may contribute to the organization and generation of ideas during the process.

5.2. Projection view

The projection view aims to provide visual validation at the overview level. To compare the results of the similarity between MVESs based on two different similarity measurements (R2), we design a tightly coupled dual view (Fig. 1 (B)) to simultaneously show different dimensionality reduction (DR) results for the same subset. Fig. 1 (B1) corresponds to the latest similarity measurement, and Fig. 1 (B2) corresponds to the other one.

DR methods are essential tools in visualization that reveal similarities and differences [VDMPVdH*09] among sequence entities. Among the commonly used DR algorithms, such as MDS [BG05], PCA, and LDA, we select the t-SNE algorithm [VdMH08] to project sequence entities for two considerations. (1) t-SNE and metric MDS can utilize the distance matrix between the entities to project them instead of the numeric feature representation that is difficult to be obtained from MVESs. (2) t-SNE is more suitable than MDS for exploring cluster structures due to the excellent separability of different clusters. We apply Procrustes transformation [GD*04] to facilitate the comparison of the two t-SNE results. Procrustes transformation is used to find the best overlap between two sets of positions by using only translation, uniform scaling, rotation, reflection, or a combination of these transformations [FCS*19]. Furthermore, based on the DR result, we further divide sequence entities into different cohorts using k-means. To achieve a reasonable clustering performance, we employ the elbow method [KM13] to find the inflection point of the sum of the squared errors (SSE) and set the position of this point as the number of clusters. The cluster number can be adjusted via Fig. 1 (b1).

The two projection views, each corresponding to a similarity measurement, show the similarity between the entities. Each entity is represented as a hollow circle, with the border color encoding its cohort. The border color of circles in the second view (Fig. 1 (B2)) is the same as in the first view (Fig. 1 (B1)), corresponding to the latest clustering result. Moreover, the two views are fully linked with each other. For example, when analysts select a subset of interest by drawing a lasso on either view, the two views are updated simultaneously, i.e., the selected circles are highlighted.

The system supports a recursive workflow for drilling down into the subset of interest for further exploration (R5). Analysts can also filter cohorts via the checkbox (Fig. 1 (c2)) or select entities freely via the lasso tool. If necessary, analysts can reverse or forward the data drill-down steps via Fig. 1 (a2).

Justification. Adding the clustering result to the projection views can improve scalability as datasets continue to grow in size [EZZ04]. The clustering technique aggregates a significant number of entities into a limited number of cohorts, which means that the number of validation elements is reduced. In addition, t-SNE is performed first and followed by k-means on that output in this view [WCR*17]. First, the clustering algorithm can execute faster on a dataset with fewer dimensions without significant loss of information. Second, this order avoids the visual confusion caused by cluster intersections, which is helpful to capture the differences in the local structure of the data under the two measurement schemes.

5.3. Pattern View

The pattern view is designed to offer visual validation at the cohort level. To summarize and compare the cohorts in temporal and multivariate perspectives (R3), we extract temporal patterns for each cohort on different attributes and visualize the cohorts' patterns in a set of pattern views (Fig. 1 (C)). The views share the same color scheme as the projection views. Considering various data formats of event attributes, we implement and improve a set of classical temporal pattern mining algorithms accordingly to create a concise yet temporal overview for sequence data in each attribute. Two examples with numeric and categorical attributes are as follows.

For each numeric attribute, our system employs the sequence alignment technique introduced in ET2 [GJG*18] to detect the temporal trends of the cohorts. That is, the sequences of each cohort are aligned temporally to a mean-sequence using DTW by multiple iterations. To measure the uncertainty within the cohort, we additionally record the set of values corresponding to each position in

the mean-sequence, and the upper and lower quartiles ($Q_{1/4}$, $Q_{2/4}$ and $Q_{3/4}$) during the iterations. The quartile trend is regarded as the temporal pattern of a cohort and is encoded as a curve boxplot [MWK14]. The multi-series line chart is designed to show the temporal trend differences of the patterns. Each median-sequence is represented as a line whose background area is used to encode the deviation information of the cohort. Usually, cohorts with wide background areas have high uncertainty and vice versa.

For each categorical attribute, we apply the MinDL+LSH algorithm proposed by Chen et al. [CXR17] to extract an optimal set of sequential patterns for each cohort. To reduce visual clutter, we only retain the patterns with *support* (the number of sequences represented by a pattern) greater than a certain threshold (0.2). In this pattern view, all sequential patterns of cohorts are displayed in a list and sorted internally by the *support*. Each row represents one sequential pattern whose height encodes the *support*. The events of each pattern are visualized as rectangles, arranged from left to right. Similarly, cohorts with many patterns have high uncertainty and vice versa.

Justification. So far, three main visualization strategies are available for MVES data, each with an example. Firstly, Wu et al. [WGW*20] used glyphs to display multiple attributes simultaneously in one view in a more compact manner. However, the massive information leads to visual clutter if too many attributes are represented. Secondly, EventPad [CvW17,CMEVW18] rewrote MVESs to high-level pattern sequences, namely single-attribute sequences. For our validation task, however, the rewriting strategy may lead to loss of information, which reduces the credibility of the conclusion. Thirdly, for multidimensional data, TPFlow [LXR18] displayed the patterns of different subsets along each dimension with multiple coordinated views, allowing users to easily observe and compare the subsets without worrying about exhausting visual channels. Therefore, we adopted the third strategy, which can be directly extended to MVES data.

5.4. Instance View

The instance view (Fig. 1 (D)) is targeted to provide visual validation at the individual level (R4). When analysts spotted instances of interest, they would toggle to the view via Fig. 1 (c1) for understanding and comparing their raw temporal information. The system supports two ways of selecting individuals, namely, click and lasso interaction. In the projection view, users can select the points of interest (e.g., outlier points, cohort intersections and edge points) by clicking on them or pick a representative individual (i.e., the group medoid) from a group with the lasso tool (see Fig. 6 (A)). These instances are represented as solid circles and their raw information on each attribute is displayed according to its data format (see Fig. 6 (B)). For each categorical attribute, a discrete event sequence is displayed in each equal-height row, where a rectangle represents an event (see Fig. 6 (B2)). For each numeric attribute, we use a multi-series line chart without background (Fig. 6 (B1)) to show the temporal trend of these instances. The view shares the same color scheme as the pattern view.

Justification. As with the pattern view, we present the original MVESs of the instances using the multiple coordinated views strategy. At the same time, to minimize the learning burden for the user, we maintain the design consistency with the pattern view. In addition, abstracting the entire cluster by the group medoid significantly reduces the number of validation instances, thereby improving the scalability of the instance-level validation.

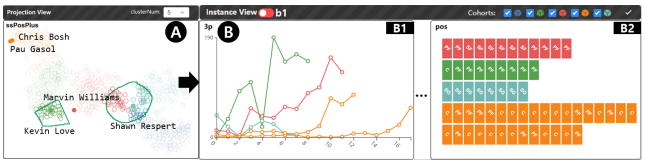

Figure 6: *The instance view (right) shows and compares the raw temporal information of the instances selected by lassoing or clicking in the projection view (left).*

6. Evaluation

We illustrate the power of the system through a case study and a user study with real-world datasets including career statistics and hospital treatments.

6.1. Example Usage Scenario

The usage scenario was conducted by an exploratory research group consisting of an analyst and his client, a basketball scout. The NBA career stats data contains season statistics for more than 2,000 NBA players from 1980 and 2017. Each player's performances per season are considered as a multivariate event, consisting of 46 attributes, such as *Pos* (Position), *Tm* (Team), *G* (Games), *3P* (3-Point Field Goals), and *BLK* (Blocks). Each player's career is regarded as an MVES, with an average length of 5.4 and a max length of 26. Navigating to the huge player stats would be overwhelming, so our system supports the free exploration of this data according to the scout's interests. First, the analyst can use the builder to quickly build player similarity measurements based on user interest. Then, the scout can complete three high-level tasks via three visual validations, respectively. (1) Redefine player classifications or find replacements for a player via the projection view. (2) Identify temporal trends of player cohorts in different statistics to direct player transactions via the pattern view. For instance, a team should buy players before their vital statistics grow rapidly and sell them before their form declines rapidly. (3) Discover desired player templates to guide player draft picks via the instance view.

To begin with, the research group browsed the distribution of the attributes and then decided to explore whether distinct temporal trends in players on the offensive end existed. Therefore, the analyst first dragged attribute blocks *2P*, *3P*, and *FT* (Free Throws) from the attribute view (Fig. 1 (A1)) onto the canvas (Fig. 1 (A3)). Next, considering the large number of current players, the analyst dragged the *Euclidean Distance* operation block with low computational complexity onto the canvas to build three univariate sequence similarities (i.e., *ss3P*, *ss2P*, and *ssFT*). Last, they used the *Weighted Summation* operation block to combine the three sequence similarities into one holistic similarity measurement named *ssOFFENCE* (Fig. 7 (A1)). Accordingly, the system yielded a projection view (Fig. 7 (A2) and colored the circles according to the clustering result. As shown in Fig. 7 (A3-A6), the red cohort maintained high levels of *3P* and *3P%* (3-Point Field Goal Percentage) throughout its player careers, while the *2P* and *FT* levels are rela-

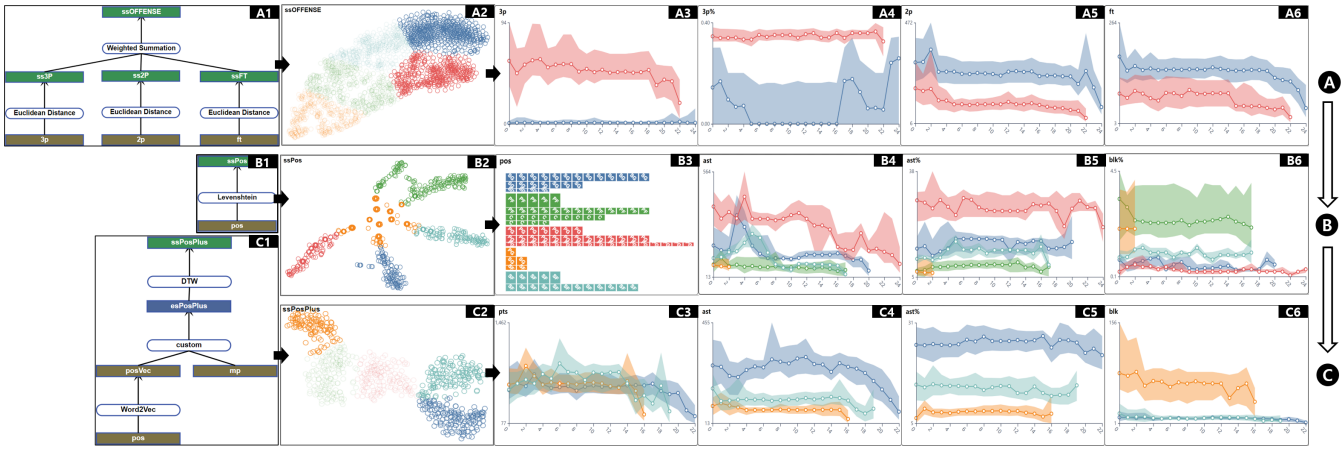

Figure 7: *Recursive exploration of NBA career statistics in three rounds. (A) Based on the offensive similarity (A1), excellent offensive cohorts are identified, understood, and focused (A2-A6). (B) The orange cluster in B2 with a remarkably short career is filtered. (C) A multivariate sequence similarity considering both MP and semantic position (Pos) is built to identify players with similar career role changes. SG, PG and C cohorts' performances are compared on multiple attributes (C3-C6).*

tively low. However, the blue cohort had excellent *2P* and *FT* performances (i.e., most shots and highest hitting percentage), which declined dramatically towards the end of their careers. The wide background area of the blue line (see Fig. 7 (A4)) implies that this cohort had a high uncertainty of *3P%* performance. To further understand the two cohorts, the scout focused on them to open a new round of exploration (Fig. 7 (B)).

After a group discussion, they planned to further build a similarity measurement to identify player cohorts with similar position/role changes (e.g., SG (Shooting Guard) → PG (Point Guard) → C (Center)) during their careers. The *Pos* attribute and *Levenshtein* [L*66] operation blocks were dragged and dropped sequentially onto the canvas to build the edit distance between the position sequences (see Fig. 7 (B1)). As shown in the projection view (Fig. 7 (B2)), the players are divided into five clear cohorts, where the orange cohort corresponds to the players with extremely short careers (see Fig. 7 (B3-B6)). Thus, the orange cohort was filtered out as the scout focused on players with long careers.

However, while many players have the same position sequence, their playing time at each position is quite different. Thus, for building a more reasonable position sequence similarity, the scout suggested considering *MP* (minutes played) instead of relying only on *Pos*. In addition, the scout emphasized that the similarities between positions should be quantified and not simply described as the same or different. For example, the similarity between SG and PG is higher than that between SG and C. Consequently, the analyst improved similarity measurement *ssPos* following three key steps (see Fig. 1 (A3)):

(1) Derive new attribute. A trained Word2Vec model was imported as an operation block, which converts each position into a vector representation. Then the *Pos* attribute and *Word2Vec* operation blocks were dragged onto the canvas to generate a new attribute named *posVec*, representing the embedding vector of position (see Fig. 1 (A3 ①)).

(2) Build event similarity. On the basis of the *posVec* and *MP* attribute blocks, the analyst used the *custom* operation block to visually define the similarity between a pair of multivariate partial-events $e^x[A^*]$ and $e^y[A^*]$ with attribute subset $A^* = \{posVec, MP\}$. The transformation formula from attributes to event similarity was defined as:

$$esPosPlus = CD\left(v^x_{posVex}, v^y_{posVex}\right) * Min\left(v^x_{MP}, v^y_{MP}\right) + \left|v^x_{MP} - v^y_{MP}\right|$$

where v^x_{mp} indicates the value of attribute *MP* of event e^x, v^y_{posVec} denotes the value of attribute *posVec* of event e^y, and *CD* is the abbreviation for Cosine Distance. The event similarity named *esPosPlus* consists of two parts: the position semantic and the *MP* distance. The formula was programmed via the formula builder panel (Fig. 4) in the second step (Fig. 1 (A3 ②)).

(3) Build sequence similarity. Based on *esPosPlus* block, a pair of sequences were aligned temporally using the *DTW* operation block, and then sequence similarity named *ssPosPlus* was calculated (see Fig. 1 (A3 ③)).

From the projection views (Fig. 1 (B)), both the *ssPos* and *ssPosPlus* similarity measurements can identify five significant cohorts corresponding to the five common positions. However, the research group preferred *ssPosPlus* because the RD result of the five clusters calculated based on it is more reasonable than *ssPos*. Specifically, the SG cohort (cyan) in Fig. 1 (B2) is far away from the other cohorts. However, it is near to the PG cohort (blue) in Fig. 1 (B1). Furthermore, because the *MP* attribute was considered, the entities within the clusters are more differentiated rather than heavily overlapping. The scout selected three cohorts (C, SG, and PG) and turned to the pattern view (Fig. 1 (C) and Fig. 7 (C3-C6)) for detailed information. Other findings are shown as follows: (1) In the second half of the C cohort's career (orange), *3P%* improved considerably, but with high uncertainty. (2) The three cohorts performed equally well in *WS/48* (Win Shares Per 48 Minutes) and *PTS* (Points), increasing and then decreasing at a high level. (3) The SG cohort gradually slipped in *AST* (Assists) but remained maintained a high level in *AST%* (Assist Percentage). For the instance-level validation, the analyst can also turn to the instance view for detailed information (see Fig. 6).

6.2. User Study

We conducted a two-part user study with twelve data analysts using the system to complete four specified similarity measurements development tasks and four visual validation tasks. These two parts corresponded to two goals: (1) Quantitatively evaluating the efficiency of the similarity builder in developing similarity measurements. (2) Qualitatively assessing the usefulness of multi-granularity visual validations for measurement schemes.

Data. We chose a publicly accessible critical care dataset, MIMIC-IV [JBP*21], which none of the participants had explored before. The dataset contains comprehensive information (e.g., laboratory measurements, medications administered, and vital signs documented) for 382,278 patients in the hospital. We selected a group of 196 patients diagnosed with major depressive disorder and retrieved their EMARs during their first hospitalization. The EMARs of each patient are processed as an MVES. Each medication administration to a patient is a multivariate event consisting of 21 attributes, such as medication, type, dose due, route, care unit, and timestamp.

Tasks. For Part 1, because no competing software can visually develop similarity measurements, we compared the task completion time between the builder and the original development manner (i.e., Python programming) to evaluate the efficiency of the builder. For Part 2, the visual validation process includes a series of complex exploration tasks, which is not a simple yes or no question. Therefore, we conducted a qualitative user study instead of a controlled quantitative experiment. Detailed tasks are available in the supplemental material.

Tasks of Part 1 (T1–T4): Developing four specified similarity measurements using the builder and programming, respectively.

Tasks of Part 2 (T5–T8): Validating the similarity measurements of T1-T4 respectively by the projection view, pattern view and instance view.

Participants. We recruited twelve participants at collaborating company with two different degrees of experience exploring event sequences using basic machine learning methods. Four analysts of them (referred to as A1–A4) are our collaborators who have worked with us from the initial stage and have extensive experience using the system for sequence data exploration. The other eight analysts, half senior analysts (A5–A8) and half junior analysts (A9–A12), have never seen the system before. Junior analysts (work experience <1 year each) have basic sequence mining algorithms and programming skills using Python, whereas senior analysts (work experience 5-8 years each) are significantly more capable than the former. All participants had no prior knowledge about the MIMIC-IV dataset. They were divided into three groups (G1–G3) based on their familiarity with the system and skill level.

Procedure. The study was conducted at the experimenter's office cubicle, with one participant at a time using the system on the experimenter's computer. We first briefly introduced our system's visual components, interaction designs and exploration workflow. Then, we taught the participants how to use the system through the usage scenario of NBA Career Statistics (Section 6.1). The tutorial lasted approximately 30 minutes.

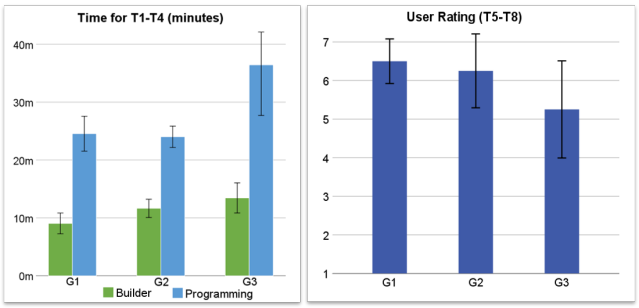

Figure 8: *Average development time for each group to complete T1–T4 using the builder and programming manner, respectively (left). Average user rating for the usefulness of the visual validations by each group after completing T5–T8.*

After a 15-minute break, we started the first part of the experiment. The participants were asked to complete four similarity measurements development tasks (T1–T4) using the builder and programming manner (Python) respectively. The order of development manners was structurally alternated in each group. All algorithms integrated in our system are also available for the programming manner. We recorded the time spent by each participant on the tasks using each implementation manner.

After a 20-minute break, we continued with the second part of the experiment. The participants were asked to validate the measurement schemes of T1-T4 respectively using the three validation views. After finishing all the four tasks, we asked the participants to rate the usefulness of the visual validations using a 7-point Likert scale from strongly disagree (1) to strongly agree (7). The validation phase was limited to two hours.

Finally, we further gathered analysts' feedback after they completed all tasks using the system.

6.2.1. Results

Below we report significant results of our quantitative and qualitative analysis. For the detailed results of twelve participants, please refer to the supplemental material.

Efficiency of the similarity builder. Comparing the task completion times (T1–T4) of the two development manners for each group, the result suggests that the builder improves the speed and efficiency of developing similarity measurements for all the analysts, especially junior analysts. As depicted in Fig. 8 (left), the similarity builder reduces the user approximately two-thirds of the time cost in developing MVES similarity measurements relative to programming. Moreover, the completion times of the tasks using the builder do not differ significantly among the three groups of participants. The completion time of G1 is slightly lower than the other two groups, indicating that the builder was friendly and accessible for users who had never used our system before.

Usefulness of the validation views. The majority of the analysts positively assessed the usefulness of the three validation views. As depicted in Fig. 8 (right), the senior analysts (G1 and G2) gave ratings above 6 (agree). However, junior analysts (G3) rated the usefulness of the validation views significantly lower than the other two groups, at about 5 (somewhat agree). It can be interpreted as

underutilization of the three validation views. The senior analysts realized the low cost of building and validating similarity measurements using the system and tried to build and validate more measurement schemes to gain optimal schemes and comprehensive insights. The junior analysts generally missed this tip.

6.2.2. User feedback

During the experiment, we received a wealth of insightful feedback and comments from analysts, which were summarized by group.

G1: Our collaborators (G1) who are very familiar with the system focus on the design justification and scalability. The majority of the analysts stated that our system's visualization and interaction designs are well-designed. However, A2 pointed out that the scrolling mechanisms used for both the builder view (for attribute distributions) and the pattern view could become ineffective if too many visual elements are represented, asking for added visual strategies (e.g., focus + context or multi-scale views).

G2: G2's senior analysts using this system for the first time praised our system because it enables analysts to explore MVES data paired with domain experts. They also appreciated that the system provided multi-granularity validation feedback for similarity measurements. "*Developing a similarity measurement is an iterative trial-and-error process.*" A6 commented, "*The similarity builder and multi-granularity visual validations did not reduce the number of iterations, but the system can reduce the time per iteration.*" Meanwhile, they offered three constructive suggestions for expanding the functionality of the system. Firstly, A5 and A8 suggested that the system should support multi-analyst collaboration to explore MVES data. Secondly, A5, A6 and A8 proposed to introduce effective visual clues to guide the construction of similarity. They commented that "*The space of the similarity measurement of MVES data determined by different choices of data subset, attributes, algorithms and parameters is vast, so we need an effective navigation.*" Thirdly, A6 and A7 mentioned that it is not enough to manage the exploration history just through the forward and back buttons, and they suggested using a tree view to handle exploration branches.

G3: Compared to G2, the junior analysts (G3) were particularly impressed by the visual programming capability of the builder. A9 and A10 commented that "*It's minimalistic yet expressive. The builder eases the development of a similarity measurement for MVES data by configuring it as human-data interaction, which reduces the programming requirements.*" Meanwhile, they desired some quantitative metrics to assess the performance of similarity measures. When the performance of the similarity measurement declines, the analyst can revert the measurement.

7. Discussion and Future Work

Generalizability. As a knowledge-assisted visual analytics prototype, our system is designed to assist analysts in exploring MVESs in various domains (e.g., electronic health records, hit sequences in racquet sports, and page logs on websites). Meanwhile, our work can also be easily extended to a wider range of temporal data, such as multivariate time-series data. For instance, air quality data can be regarded as a special case of MVES data whose event attributes (e.g., PM2.5, PM10, SO2, NO2, O3, and CO) are all in numeric data format. For exploring such data, our system can be directly utilized to build purpose-specific site similarity measurements, validate their performance, and focus on areas of interest for further exploration.

Scalability. Many common classical algorithms are available for conveniently building similarity measurements. These algorithms affect the scalability of our system to varying degrees. Nevertheless, the scalability issues about "algorithm" can be addressed in the following perspectives. Firstly, the system allows users to import a more efficient algorithm as a new operation block via an operation import panel. Secondly, similar to the method of Stolper et al. [SPG14], we will employ progressive visual analytics [MSA*19, ASSS18] solution to visualize meaningful partial results during execution for analyst intervention without waiting for the computation to complete. Several scalability issues also exist in our visualization views. The system supports visual validations of similarity measurements at three granularities, which is a promising solution for improving the scalability about "entities". Although we try to use multi-view visualization for supporting the exploration of more attributes, we still encounter scalability issues about "attributes". Browsing information on multiple attributes at the same time is overwhelming for the user, which means that the scrolling mechanism is ineffective. Focus + context or multi-scale strategies are useful solutions.

Limitations. We identified two limitations in our work. Firstly, the system cannot fully handle the diversity of data formats of event attributes. While the builder supports event attributes in various data formats to develop MVES similarity measurements, the visualization components support only the two most common types of event attributes (i.e., category, and numeric) for visual validations of similarity measurements. However, the data formats of event attributes of real-world MVES data include not only category and numeric, but also range, string, vector, set, etc. Therefore, more summarization techniques for temporal data in different data formats need to be integrated into the system to support more comprehensive visual validations. Secondly, the system is powerful but not smart. More visual cues are needed to navigate the vast space of the similarity measurements. Analysts use our system to explore MVES data, relying heavily on their experience. We plan to collect analyst behavior data as the basis for intelligent guidance.

8. Conclusion

In this paper, we introduce a visual analytics system for similarity-based exploration of MVESs by building and validating the similarity measurements. We generalize a unified process for calculating MVES similarity and specify the system requirements for exploring such data. The system features the MVES similarity builder, a visual programming tool that eases, through the assembly of visual building blocks, the process of development of similarity measurements for MVES data. Built upon the builder, we further propose a visual analytics system that provides visual validations at three granularities (i.e., overview, cohort, instance) for similarity measurements. The framework also supports a recursive workflow for drilling down into the subset of interest. We illustrate the usefulness and efficiency of our system through a usage scenario and a user study with real-world MVES data from the sports and medical domains.

9. Acknowledgements

We would like to thank all participants in our study for their availability and significant comments. This work is a part of the research supported by National Natural Science Foundation of China (61872164).

References

[ASSS18] ANGELINI M., SANTUCCI G., SCHUMANN H., SCHULZ H.-J.: A review and characterization of progressive visual analytics. In *Informatics* (2018), vol. 5, Multidisciplinary Digital Publishing Institute, p. 31. 10

[BG05] BORG I., GROENEN P. J.: *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005. 6

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE transactions on visualization and computer graphics 17*, 12 (2011), 2301–2309. 5

[BSH*15] BACH B., SHI C., HEULOT N., MADHYASTHA T., GRABOWSKI T., DRAGICEVIC P.: Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 559–568. 3

[BWS*12] BERNARD J., WILHELM N., SCHERER M., MAY T., SCHRECK T.: Timeseriespaths: Projection-based explorative analysis of multivariate time series data. 3

[CD18] CAVALLO M., DEMIRALP Ç.: Clustrophile 2: Guided visual clustering analysis. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 267–276. 2

[CMEVW18] CAPPERS B. C., MEESSEN P. N., ETALLE S., VAN WIJK J. J.: Eventpad: Rapid malware analysis and reverse engineering using visual analytics. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)* (2018), IEEE, pp. 1–8. 3, 6, 7

[CvW17] CAPPERS B. C., VAN WIJK J. J.: Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 532–541. 3, 6, 7

[CXR17] CHEN Y., XU P., REN L.: Sequence synopsis: Optimize visual summary of temporal event data. *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 45–55. 7

[CYP*18] CHEN Q., YUE X., PLANTAZ X., CHEN Y., SHI C., PONG T.-C., QU H.: Viseq: Visual analytics of learning sequence in massive open online courses. *IEEE transactions on visualization and computer graphics 26*, 3 (2018), 1622–1636. 2

[DPSS16] DU F., PLAISANT C., SPRING N., SHNEIDERMAN B.: Eventaction: Visual analytics for temporal event sequence recommendation. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2016), IEEE, pp. 61–70. 2

[EMK*19] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA N. S., TELEA A. C.: Toward a quantitative survey of dimension reduction techniques. *IEEE transactions on visualization and computer graphics 27*, 3 (2019), 2153–2173. 2

[EZZ04] EICK C. F., ZEIDAT N., ZHAO Z.: Supervised clustering-algorithms and benefits. In *16Th IEEE international conference on tools with artificial intelligence* (2004), IEEE, pp. 774–776. 6

[FCS*19] FUJIWARA T., CHOU J.-K., SHILPIKA S., XU P., REN L., MA K.-L.: An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE transactions on visualization and computer graphics 26*, 1 (2019), 418–428. 6

[GD*04] GOWER J. C., DIJKSTERHUIS G. B., ET AL.: *Procrustes problems*, vol. 30. Oxford University Press on Demand, 2004. 6

[GFL*20] GUO R., FUJIWARA T., LI Y., LIMA K. M., SEN S., TRAN N. K., MA K.-L.: Comparative visual analytics for assessing medical records with sequence embedding. *Visual Informatics 4*, 2 (2020), 72–85. 2, 3

[GGJ*21] GUO Y., GUO S., JIN Z., KAUL S., GOTZ D., CAO N.: Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics* (2021). 2

[GJG*18] GUO S., JIN Z., GOTZ D., DU F., ZHA H., CAO N.: Visual progression analysis of event sequence data. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 417–426. 2, 6

[GLW21] GE T., LEE B., WANG Y.: Cast: Authoring data-driven chart animations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–15. 3

[JBP*21] JOHNSON A., BULGARELLI L., POLLARD T., HORNG S., CELI L. A., MARK R.: Mimic-iv (version 1.0). *PhysioNet* (2021). 9

[JCG*20] JIN Z., CUI S., GUO S., GOTZ D., SUN J., CAO N.: Carepre: An intelligent clinical decision assistance system. *ACM Transactions on Computing for Healthcare 1*, 1 (2020), 1–20. 2

[JME10] JAVED W., MCDONNEL B., ELMQVIST N.: Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics 16*, 6 (2010), 927–934. 3

[JSS*19] JÖNSSON D., STENETEG P., SUNDÉN E., ENGLUND R., KOTTRAVEL S., FALK M., YNNERMAN A., HOTZ I., ROPINSKI T.: Inviwo—a visualization system with usage abstraction levels. *IEEE transactions on visualization and computer graphics 26*, 11 (2019), 3241–3254. 3

[KM13] KODINARIYA T. M., MAKWANA P. R.: Review on determining number of cluster in k-means clustering. *International Journal 1*, 6 (2013), 90–95. 6

[KR05] KEOGH E., RATANAMAHATANA C. A.: Exact indexing of dynamic time warping. *Knowledge and information systems 7*, 3 (2005), 358–386. 4

[L*66] LEVENSHTEIN V. I., ET AL.: Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (1966), vol. 10, Soviet Union, pp. 707–710. 8

[LLMB18] LAW P.-M., LIU Z., MALIK S., BASOLE R. C.: Maqui: Interweaving queries and pattern mining for recursive event sequence exploration. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 396–406. 2

[LLX*10] LIU Y., LI Z., XIONG H., GAO X., WU J.: Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining* (2010), IEEE, pp. 911–916. 2

[LPK*15] LOORAK M. H., PERIN C., KAMAL N., HILL M., CARPENDALE S.: Timespan: Using visualization to explore temporal multidimensional data of stroke patients. *IEEE transactions on visualization and computer graphics 22*, 1 (2015), 409–418. 3

[LXR18] LIU D., XU P., REN L.: Tpflow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 1–11. 7

[MeV] MEVIS MEDICAL SOLUTIONS AG: Mevislab. Accessed: 2022-02-17. URL: https://www.mevislab.de. 3

[MR97] MANNILA H., RONKAINEN P.: Similarity of event sequences. In *Proceedings of TIME'97: 4th International Workshop on Temporal Representation and Reasoning* (1997), IEEE, pp. 136–139. 2

[MSA*19] MICALLEF L., SCHULZ H.-J., ANGELINI M., AUPETIT M., CHANG R., KOHLHAMMER J., PERER A., SANTUCCI G.: The human user in progressive visual analytics. In *Eurovis (short papers)* (2019), pp. 19–23. 10

[MWK14] MIRZARGAR M., WHITAKER R. T., KIRBY R. M.: Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 2654–2663. 7

[SC07] SALVADOR S., CHAN P.: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis 11*, 5 (2007), 561–580. 5

[SPG14] STOLPER C. D., PERER A., GOTZ D.: Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1653–1662. 10

[TLS21] THOMPSON J. R., LIU Z., STASKO J.: Data animator: Authoring expressive animated data graphics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–18. 3

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 6

[VDMPVdH*09] VAN DER MAATEN L., POSTMA E., VAN DEN HERIK J., ET AL.: Dimensionality reduction: a comparative. *J Mach Learn Res 10*, 66-71 (2009), 13. 6

[WCR*17] WENSKOVITCH J., CRANDELL I., RAMAKRISHNAN N., HOUSE L., NORTH C.: Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 131–141. 6

[WGW*20] WU J., GUO Z., WANG Z., XU Q., WU Y.: Visual analytics of multivariate event sequence data in racquet sports. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2020), IEEE, pp. 36–47. 2, 3, 7

[WLG*21] WU J., LIU D., GUO Z., XU Q., WU Y.: Tacticflow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics* (2021). 3

[WPTMS12] WONGSUPHASAWAT K., PLAISANT C., TAIEB-MAIMON M., SHNEIDERMAN B.: Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with computers 24*, 2 (2012), 55–68. 2, 3

[WRF*11] WASER J., RIBICIC H., FUCHS R., HIRSCH C., SCHINDLER B., BLOSCHL G., GROLLER E.: Nodes on ropes: A comprehensive data and control flow for steering ensemble simulations. *IEEE transactions on visualization and computer graphics 17*, 12 (2011), 1872–1881. 3

[WS09] WONGSUPHASAWAT K., SHNEIDERMAN B.: Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *2009 IEEE Symposium on Visual Analytics Science and Technology* (2009), IEEE, pp. 27–34. 2, 3

[ZDFD15] ZGRAGGEN E., DRUCKER S. M., FISHER D., DELINE R.: (sl qu) eries: Visual regular expressions for querying and exploring event sequences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 2683–2692. 3