EventAction: A Visual Analytics Approach to Explainable Recommendation for Event Sequences

FAN DU, CATHERINE PLAISANT, NEIL SPRING, KENYON CROWLEY, and BEN SHNEIDERMAN, University of Maryland, USA

People use recommender systems to improve their decisions, for example, item recommender systems help them find films to watch or books to buy. Despite the ubiquity of item recommender systems, they can be improved by giving users greater transparency and control. This paper develops and assesses interactive strategies for transparency and control, as applied to event sequence recommender systems, which provide guidance in critical life choices such as medical treatments, careers decisions, and educational course selections. This paper's main contribution is the use of both record attributes and temporal event information as features to identify similar records and provide appropriate recommendations. While traditional item recommendations are based on choices by people with similar attributes, such as those who looked at this product or watched this movie, our event sequence recommendation approach allows users to select records that share similar attribute values and start with a similar event sequence. Then users see how different choices of actions and the orders and times between them might lead to users' desired outcomes. This paper applies a visual analytics approach to present and explain recommendations of event sequences. It presents a workflow for event sequence recommendation that is implemented in EventAction and reports on three case studies in two domains to illustrate the use of generating event sequence recommendations based on personal histories. It also offers design guidelines for the construction of user interfaces for event sequence recommendation and discusses ethical issues in dealing with personal histories. A demo video of EventAction is available at https://hcil.umd.edu/eventaction.

CCS Concepts: • **Human-centered computing → Visual analytics**; *Graphical user interfaces*; • **Information systems → Recommender systems**; *Similarity measures.*

Additional Key Words and Phrases: similarity, personal record, multidimensional data visualization, temporal visualization, decision making, visual analytics

ACM Reference Format:
Fan Du, Catherine Plaisant, Neil Spring, Kenyon Crowley, and Ben Shneiderman. 2019. EventAction: A Visual Analytics Approach to Explainable Recommendation for Event Sequences. *ACM Trans. Interact. Intell. Syst.* 1, 1 (February 2019), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems are widely used to assist people in making decisions, for example, item recommender systems help customers to find films to watch or books to buy. Despite the ubiquity of item recommender systems, they can be improved by giving users greater transparency and control. This paper develops and assesses interactive strategies for transparency and control, as applied to event sequence recommender systems, which can provide guidance in critical life

Authors' address: Fan Du, fan@cs.umd.edu; Catherine Plaisant, plaisant@cs.umd.edu; Neil Spring, nspring@cs.umd.edu; Kenyon Crowley, kcrowley@rhsmith.umd.edu; Ben Shneiderman, ben@cs.umd.edu, University of Maryland, College Park, MD, 20742, USA.

2160-6455/2019/2-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

choices such as medical treatments, careers decisions, and educational course selections. Time-stamped event sequence data has become ubiquitous with the development of mobile devices, electronic communication, and sensor networks. It can be collected from social network activities, online clickstreams, electronic health records, and student academic activities. Event sequence recommender systems use archives of similar event sequences, such as patient histories or student academic records, to give users insight into the order and timing of their choices, which are more likely to lead to their desired outcomes.

Imagine the following scenario: I am a student at the end of my second year of graduate school. I wish to become a professor and wonder what jobs other students like me got. Then, I wonder what those who ended up being professors did in their last two years of studies. Did they go on internships? When and how many times? I know that publishing is important, but when did they typically publish papers? Does it seem better to start early or all at the end? Did they get a masters on the way? Did they work as teaching assistants? Early on or later toward the end? So I meet with my department's graduate advisor. He pulls a set of students' records from the campus archives who are similar to me based on their first two years of studies. He explains to me their outcomes in terms of the time it took to graduate and job type. Then, the advisor looks at those who became professors, we review the recommendations together and discuss an action plan - combining the wisdom of the advisor and the system's recommendations based on events and the orders and times between them identified as correlated with becoming a professor.

1.1 Problem and Approach

The research question addressed in this work is: *What combination of algorithmic analysis and interactive visual exploration can augment analysts' ability to find similar records, review recommended actions, and make action plans to improve outcomes?*

To find a group of records with features in common with a seed record, one approach is to specify a query and the results are records that exactly match the query rules. Extensions to standard query languages (e.g., TQuel [69] and T-SPARQL [25]) have been introduced to ease the task of querying temporal data. Such temporal queries typically consist of elements such as the required events, temporal relationships between the events, and attribute ranges of the events or records.

The temporal query approach is useful when users have prior assumptions about the data so as to specify query rules. However, it is unsuitable to be applied alone for the task of finding similar records—only a few or zero results will be found if many query rules are specified to fully characterize the seed record; or if only a few rules are used, the results may be quite dissimilar to the seed record in aspects outside the query rules. Besides, precisely formulating temporal queries remains difficult and time-consuming for many domain experts. Our approach enables users to find and explore similar records using both record attributes and temporal event information as similarity criteria. To encourage engagement and inspire users' trust in the results, it provides different levels of controls and context for users to adjust the similarity criteria.

Understanding how different sequences of events lead to different outcomes is an important task in event sequence analysis, leading to hypotheses about causation. For example, OutFlow [82] uses a network structure to aggregate similar event sequences into progression pathways and summarizes the pathways' possible outcomes. TreatmentExplorer [20] provides a novel graphical interface for presenting the outcomes, symptoms, and side effects of treatment plans. CoCo [43] helps analysts compare two groups of records (e.g., with different outcomes) and uses high-volume hypothesis testing to systematically explore differences in the composition of the event sequences found in the two groups.

These tools visualize the outcomes of a given set of records, enabling users to see the outcomes and progression pathways associated with these records. Our approach is to extend these work

by providing recommended sequences of temporal events that might help achieve users' desired outcomes. It also allows users to define personalized action plans and provides feedback on the probability of success. In addition, while most existing tools assume a binary outcome, our approach enables users to explore multiple outcomes.

1.2 Contributions

In this work, we introduce EventAction, an interactive prescriptive analytics system and user interface to assist users in making action plans and to raise users' confidence in the action plans. The main contribution of this work is the use of both record attributes and temporal event information as features to identify similar records and provide appropriate recommendations. While traditional item recommendations are generated based on choices by people with similar attributes, such as those who looked at this product or watched this movie, the event sequence recommendation approach allows users to select records that share similar attribute values and start with a similar event sequence, and then see how different choices of actions and the orders and times between them might lead to users' desired outcomes.

Our preliminary work designed interface components for finding similar records [17, 18] and reviewing recommendations of action plans [16]. This paper provides a comprehensive final review of the entire EventAction project with the following new contributions:

- A thorough literature review of existing techniques, software tools, and ethical issues related to event sequence recommendation.
- A description of the final EventAction system that revises and integrates the interface components from our preliminary work [16–18] and provides an automatic sequence recommendation algorithm to reduce users' effort in using the system. The integrated system supports a systematic analytical workflow for event sequence recommendation that will be applicable in diverse applications.
- A report on three case studies in two domains that provide evidence of the effectiveness of generating event sequence recommendations based on personal histories.
- A set of five design guidelines for the construction of event sequence recommendation user interfaces and three usage guidelines for mitigating the ethical issues in dealing with personal histories.

2 BACKGROUND AND RELATED WORK

We summarize existing techniques and software tools that can contribute to our goal of enabling users to generate recommendations of event sequences that might lead to their desired outcome. Our work is particularly inspired by previous research on recommender systems, similarity measures, event sequence analysis, and ethical issues in information systems.

2.1 Recommender Systems

When making decisions, people often lack sufficient experience or competence to evaluate the potentially overwhelming number of alternative choices. Recommender systems tackle this challenge by providing personalized suggestions for items likely to be of use to a user [60].

Recommendation Techniques. Previous work identified four major classes of recommendation techniques [10]. The two most popular ones are content-based, which recommends items similar to what the users liked in the past [12], and collaborative filtering, which finds other users with similar tastes and recommends items they liked to the current user [39, 59, 63]. When large-scale user profiles are available, demographic techniques can be used to generating user-specific recommendations based on common patterns in the population [34]. When domain knowledge

about item features are available, knowledge-based techniques can estimate how much an item meets a user's needs and identify the best matches [8, 61].

In practical applications, multiple recommendation techniques are often combined to encourage the strength and diminish the weakness [13, 49]. Besides, recent advances reveal that it is important to incorporate temporal information into the recommendation process. For example, seasons and opening hours are important context for recommending tourist locations [42] and users' daily activity patterns should be considered when recommending social events [41].

Evaluating Recommender Systems. Approaches for evaluating recommender systems differ depending on the goals of an evaluation. Early work in this field primarily focused on the accuracy of recommendation algorithms. For example, Herlocker et al. [28] used mean absolute error to measure the deviation between preference ratings predicted by algorithms and provided by users. Shardanand and Maes [66] discovered that error of the extremes can be valuable and measured separately large errors between the predicted and user ratings.

Follow-up research found accurate predictions crucial but insufficient for developing recommender systems that can actually influence the behavior of users. A variety of measures regarding user satisfaction have been introduced to fill this gap. For example, McNee et al. [48] built a citation recommender system for research papers and measured the novelty of the recommended references to users. In an experiment on music recommender systems, Sinha and Swearingen [68] examined the role of transparency by measuring recommenders' ability to explain the recommendations to users. Besides, commercial recommender systems also quantify user satisfaction with the number of product purchases and returns [22, 39, 41].

Opportunities. Our recommendation approach extends the collaborative filtering technique since we also generate recommendations by referring to archived records that share similar features with the seed record. However, compared to traditional recommender systems that recommend items such as books to read or social events to attend, our paper focus on recommending sequences of temporal events. Here, each event can be treated as an item and two additional dimensions need to be considered: (1) the combinations of events and their orders, and (2) the timings of the events. Besides, we develop a prescriptive analytics system designed to present and explain the recommendations. It augments traditional recommender systems by guiding users to define a personalized action plan associated with an increased probability of success.

2.2 Similarity Measures

Similarity is a fundamentally important concept in many research domains [2]. For example, in bioinformatics for gene sequence alignment [35] or protein clustering [38], in linguistics for approximate string matching [52] or text categorization [11], in computer vision for face recognition [62], and in healthcare for identifying similar patients [73, 81].

Multidimensional Data. Data scientists investigated how to measure the similarity between two multidimensional data cubes. For example, Baikousi et al. [5] conducted user studies to explore various distance functions to identify the preferred measurement between the values of a dimension and between data cubes. Spertus et al. [70] presented an empirical evaluation of similarity measures for recommending online communities to social network users, where the effects of the measures are determined by users' propensity to accept the recommendation. Sureka and Mirajkar [71] extensively studied different similarity measures for online user profiles and discovered that no single similarity measure could produce the best results for all users. They suggested using different similarity measure for different users.

We extend existing work on perceived similarity and study temporal data, which is an important component of people's healthcare histories, academic records, and online profiles. Our interviews confirmed that choices of similarity measures rely on users' preferences and analysis goals, and our user studies revealed that providing controls and context will increase users' engagement and trust in similarity search results.

Temporal Data. To find records of event sequences with features in common with a seed record, one approach is to specify a query and the results are records that exactly match the query rules. Extensions to standard query languages (e.g., TQuel [69] and T-SPARQL [25]) have been introduced to ease the task of querying temporal data. Temporal queries typically consist of elements such as the required events, temporal relationships between the events, and attribute ranges of the events or records. Precisely formulating temporal queries remains difficult and time-consuming for many domain experts. Visual tools have been developed to further ease the task by enabling users to interactively specify query rules and providing visual feedback to facilitate the iterative refinements of the queries (e.g., (s|qu)eries [87], COQUITO [33], and EventFlow [51]).

The temporal query approach is useful when users have a prior assumption about the data such as hypotheses or domain knowledge, so as to specify the query rules. However, it is unsuitable to be applied alone for the task of finding similar records—only a few or zero results will be found if many query rules are specified to fully characterize the seed record, or if only a few rules are used, the results may not be similar to the seed record in aspects outside the query rules.

An alternative approach to finding similar records is to start with the seed record, determine useful patterns, and search for records with similar patterns. Mannila and Ronkainen [44] presented a model for measuring the similarity of event sequences. The model computes an edit distance based on three transformation operations at the event level, including insert, delete, and move. This approach can preserve the order of the matched events and performs better when the number of operations is small. Match & Mismatch measure [85] introduces a similarity score that emphasizes the time difference of matched events and the number of mismatches, which supports matching without preserving the order. Besides, a visual interface was also provided to show a ranked list of similar records and allow users to adjust parameters. Recent work [77, 78] describes more advanced similarity measures for specific domains and problems. In addition to event sequences, techniques for finding similar records have been developed in other domains such as the similarity-based data-driven forecasting for time series [9].

Our work extends existing similarity metrics for temporal data and enables users to find and explore records that are similar to a seed record using both record attributes and temporal event information. To encourage engagement and inspire users' trust in the results, it also provides different degrees of controls and levels of context that allow users to adjust the similarity criteria.

2.3 Event Sequence Analysis

Data that contains temporal information can be modeled as sequences of temporal events, which appear in a wide range of domains, from engineering, to social media, finance, and healthcare. Techniques for representing event sequences and extracting insights from them are crucial to developing novel solutions and being increasingly studied.

Visual Representations. Starting with LifeLines [58], early research on event sequence visualization focuses on depicting the medical history of a single patient (e.g., Bade et al. [4], Harrison et al. [27], and Karam [32]). These tools allow users to visually inspect trends and patterns in a record by showing detailed events. LifeLines2 [79] extends this approach to multiple records but does not scale well when displaying a large number of records in a stacked manner.

Techniques have been introduced to handle large sets of records by offering time or category based aggregations. LifeFlow [83] introduces a method to aggregate multiple event sequences by combining them into a tree structure on an alignment point. Likewise, OutFlow [82] combines multiple event sequences based on a network of states. EventFlow [50] extends LifeFlow's concept to interval events and introduces simplification strategies to deal with large data volumes and pattern variety [19]. DecisionFlow [24] provides supports for analyzing event sequences with larger numbers of categories.

Our visualization designs are inspired by prior work and adapted to the needs of showing both detailed histories of individual records and activity summaries of groups.

Frequent Sequence Mining. One popular research topic in temporal data mining is discovering frequently occurring sequential patterns, which can generate novel insights and drive decision making [36]. Many techniques have been developed to support this task and the main challenge is that a combinatorially explosive number of intermediate subsequences need to be examined. Early work mainly focused on developing efficient and automatic algorithms. Apriori-like [1, 45] approaches assume that frequent patterns cannot contain any non-frequent sub-patterns. Given a percentage prevalence threshold, they start by collecting frequent patterns containing only one frequent event and then iteratively grow the patterns by appending new events. The process stops when no more frequent patterns can be found. These approaches become less efficient as the pattern volume or length grows.

Follow-up work addressed this issue and improved the procedure. For example, PrefixSpan [55] and SPADE [86] reduce the number of data scans, and SPAM [3] uses a bitmap representation to encode the event sequences and accelerates the mining computations with bitwise operations. Recently, Perer and Wang [57] introduced a visual interface for these black-box automatic algorithms. It enables users to explore the results of frequent sequences at different levels of details.

Frequent sequential patterns can provide guidance for users to identify important activity patterns, especially for patterns that occur frequently in archived records having the seed record's desired outcome. In our paper, we will explore frequent sequence mining techniques and apply them in the system.

Outcome Analysis. Understanding how different sequences of events lead to different outcomes is an important task in event sequence analysis, leading to hypotheses about causation. OutFlow [82] uses a network structure to aggregate similar event sequences into progression pathways and summarizes the pathways' possible outcomes. Its application for electronic medical records, CareFlow [56], allows doctors to analyze treatment plans and their outcomes for patients with certain clinical conditions. TreatmentExplorer [20] provides a novel graphical interface for presenting the outcomes, symptoms, and side effects of treatment plans. CareCruiser [26] enables doctors to retrospectively explore the effects of previously applied clinical actions on a patient's condition. CoCo [43] helps analysts compare two groups of records (e.g., with different outcomes) and uses high-volume hypothesis testing to systematically explore differences in the composition of the event sequences found in the two groups. MatrixWave [88] allows the exploration and comparison of two sets of event sequences with different outcomes by displaying the event sequences in a matrix and showing their differences at each step.

These tools visualize the outcomes of a given set of records, enabling users to see the outcomes and progression pathways associated with these records. Our approach is to extend these work by providing recommended sequences of temporal events that might help achieve users' desired outcomes. It also allows users to define personalized action plans and provides feedback on the probability of success. In addition, while most existing tools assume a binary outcome, our approach enables users to explore multiple outcomes.

2.4 Ethical Issues in Information Systems

While information technology offers powerful tools that can serve to improve people's lives, the same technology may also raise ethical issues such as threatening our privacy or providing inaccurate information that misleads our decisions. Mason [46] summarizes four types of ethical issues in information systems: privacy (what information to reveal), accuracy (who is responsible for the authenticity and accuracy), property (who owns information), and accessibility (what information can a person or an organization obtain). Similarly, Nissenbaum [53] introduces the concept of accountability in computing to ensure that harms and risks caused by technology can be answered and handled.

In our paper, by working with real users and domain professionals, we review the ethical issues in dealing with personal histories. Specifically, we discuss (1) what the potential biases are in using histories of similar others to provide recommendations, (2) what the potential dangers are in allowing advisees to use the system alone, and (3) how to balance the opinions of advisors and the recommendations generated from data, especially when there is a contradiction. We discuss these ethical issues and propose possible solutions.

3 DESCRIPTION OF THE USER INTERFACE

Starting with a current seed record for review and a set of archived records, EventAction provides controls and visualizations for finding a group of archived records that are most similar to the seed record. Each record is represented as a set of record attributes (e.g., age and major) and a sequence of events, where each event belongs to a particular event category. Outcomes are often defined by the inclusion of certain events in a record, for example, events representing students' first placements. EventAction estimates the seed record's potential outcomes based on the outcome distribution of the similar archived records, and recommends actions by summarizing the activities of those who achieved the desired outcome. Recommended action plans for the seed record can be adjusted and EventAction provides immediate feedback by showing how the plan affects the outcome estimation. This section describes the user interface and analytical workflow of EventAction[1].

3.1 Design Process

The design process of EventAction was inspired by the nine-stage framework proposed by Sedlmair et al. [64]. We chose a specific application domain (student advising) to drive a design study consisting of two stages. In the first stage, we designed an initial prototype of EventAction that uses a black-box algorithm to find similar records and allows users to specify action plans with guidance and feedback generated [16]. Then, we designed interfaces and visualizations that provide controls and context for users to interactively find and explore records that are similar to a seed record [17, 18]. Each stage roughly matches the *learn* (visualization literature), *discover* (tasks and needs), *design* (visual, interaction, and algorithm), *implement* (prototypes), *deploy* (to domain expert and gather feedback), *reflect* (on designs and refine guidelines), and *write* (design study paper) stages in that framework. In this paper, we report the final design of EventAction, which revises and integrates the visual components designed in our previous studies and an automatic sequence recommendation algorithm. We also conducted additional case studies in marketing and healthcare domains to further evaluate and refine the system prototype.

3.2 Interface Overview

The EventAction user interface consists of ten coordinated interface components, supporting a seamless analytical workflow for developing action plans to achieve the desired outcome. These

[1]EventAction is available for licensing. To request a review copy of EventAction, contact plaisant@cs.umd.edu.

components are organized into two tabs, one for finding similar records (Figure 1b) and the other for reviewing recommendations of action plans (Figure 1a). Users can switch between these tabs during the analysis.

3.3 Finding Similar Records

Seed Record Timeline. The seed record's history of activities is shown as an aggregated timeline in a timetable (Figure 2a), where each row represents an event category and each column represents a time period. Events in each table cell are aggregated and represented as a square in gray and the number of event occurrences is represented by the size of the square. Users can specify temporal patterns of the seed record on the timeline and use them as similarity criteria for the search. In Figure 2a, a temporal pattern has been specified based on the seed record's research activities (no papers in the first two years and late selection of an advisor). The temporal criteria are added as glyphs in the criteria control panel. Users can hover on a glyph to highlight the temporal pattern and the focused criterion in other visualizations in an orange color.

Similarity Criteria Controls. All available criteria are shown. Categorical criteria (such as major) and numerical criteria (such as GPA) are automatically extracted from the available data, and temporal criteria are added when specified by users. Each criterion is displayed as a rectangular glyph (Figure 2b) showing its name, the value for the seed record, and the distribution of all archived records. Distributions of archived records are shown as histograms where the height of the bars represents the numbers of records. For categorical and numerical criteria, the bars are exact values or ranges of values, respectively. For temporal criteria, the bars are ranges of pattern difference scores. Records with a zero difference score have exactly the same specified temporal pattern. Users can hover on a temporal criterion to highlight the corresponding temporal pattern in the seed record timeline. Users can select how the criterion is to be used: "Ignore" (×), allow "Close Match" (~), or require "Exact Match" (=). A tolerance range can also be defined to treat multiple categorical values or a range of numerical values as an equivalent of the value of the seed record (e.g., treat M.S. and Ph.D. equally or set a GPA range between 3.2 and 3.7). The weight of each criterion can also be adjusted. As users adjust the controls, the results are updated immediately and reflected in all visualizations. Users can reorder the criteria by dragging the glyphs. Changes in order are reflected in other interface components but do not affect which records are included in the result set.

Similarity Distribution. Based on the criteria settings, a similarity score is computed for each archived record and a histogram of the scores is displayed (Figure 2c). Users can adjust the portion of the histogram that is selected for the results, i.e., the peer group. In Figure 2c, the top 12% most

Fig. 1. An overview of the EventAction user interface. Interface components are organized into two tabs: (a) for reviewing recommendations of action plans and (b) for finding similar records. (c) The display expands progressively as users request more controls.

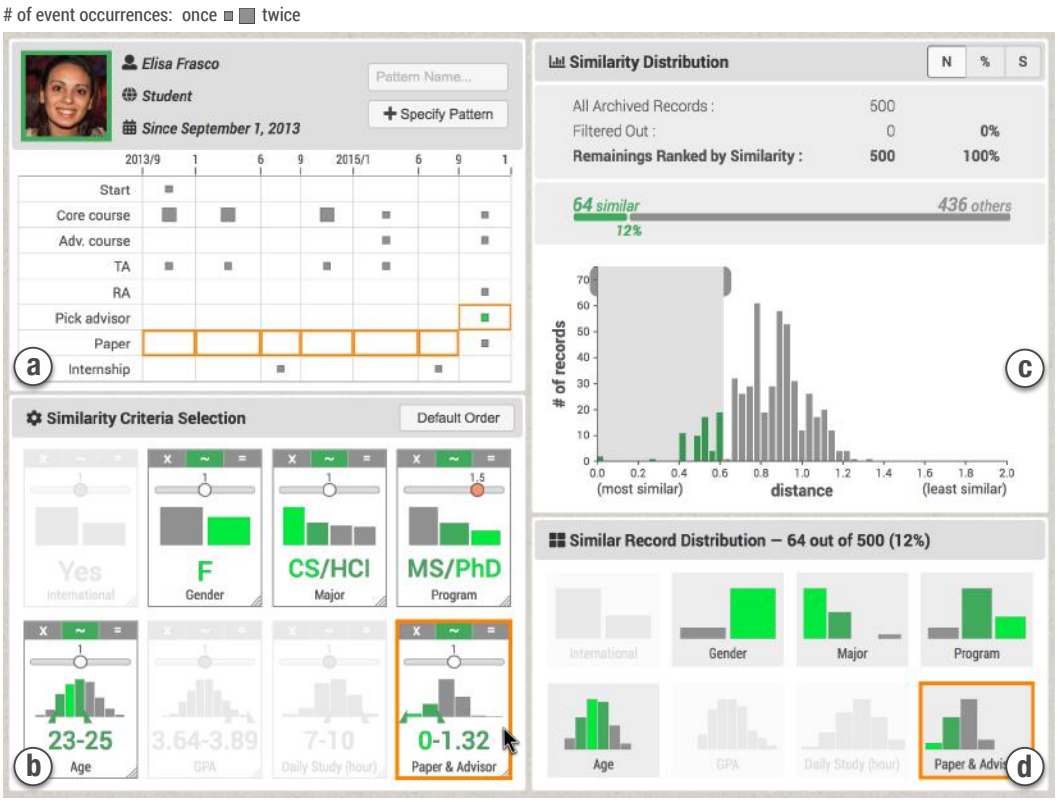

Fig. 2. Four basic components for finding similar records and adjusting the peer group: (a) seed record timeline, (b) similarity criteria controls, (c) similarity distribution, and (d) similar record distribution. In this example, a total of 8 similarity criteria are used, including one temporal criterion (Paper & Advisor). The mouse cursor is hovering on that user-defined temporal criterion representing "no papers in the first two years and late selection of an advisor." This criterion and the corresponding temporal pattern are highlighted in orange.

similar records (64 out of 500) are selected. Since the similarity scores change when users adjust the criteria controls, we provide three options to help users keep track of the record selection (shown as radio buttons in the toolbar): the "by Top N" option keeps users' selection of a fixed number of most similar records, the "by Percentage" option keeps the selection of a fixed percentage of most similar records, and the "by Similarity" option selects records whose similarity scores are above a user-defined threshold.

Similar Record Distribution. A separate view shows barchart distributions of criteria values of (only) the similar records (Figure 2d). The layout of the barcharts is consistent with the layout of the glyphs of the criteria control panel and the color of the bars is consistent with other components of the interface. Users can hover on a single bar to review the criterion range of values and number of records, and hover on a bar chart to highlight that criterion in other visualizations.

Ranked List of Similar Records. The individual records are displayed in a ranked list, showing the attribute values and the event history for each record (Figure 3d). For privacy, this panel of

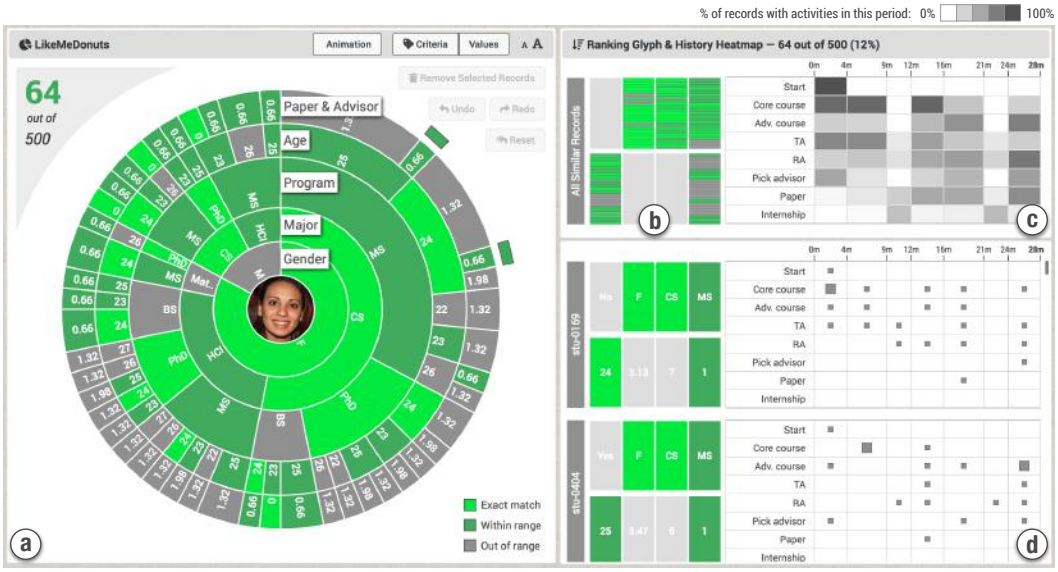

Fig. 3. Advanced visualization components for reviewing and refining peer groups: (a) LikeMeDonuts representing criteria values of the similar records as a hierarchical tree, (b) Ranking Glyph providing a compact overview of the similar records ranked by similarity, (c) History Heatmap showing the popularity of the temporal events among similar records, and (d) ranked list of similar records, displaying detailed information of individual records.

individual records will need to be hidden when users do not have proper permission. Part of the overviews or the labels may also need to be hidden if the number of records included is too low.

LikeMeDonuts. LikeMeDonuts is a radial space-filling visualization that shows the criteria values of the similar records as a hierarchical tree (Figure 3a). An image of the seed record is placed at the center, anchoring the display on that person. Each donut ring represents a criterion (and one level of a tree structure). Criteria set to "Ignore" in the similarity criteria controls are not displayed. Ring sectors in bright green represent the proportion of people in the group whose values exactly match the value of the seed record, sectors in dark green represent those within the user-specified tolerance ranges, and gray sectors represent those outside tolerance ranges.

A thin additional partial ring is shown outside the donuts to highlight the records that are most similar to the seed record (based on the selected criteria). The arc is in bright green if the record's criteria values are all exactly matched, or in dark green if all criteria values are within range. When integrated into the larger interface, in Figure 3a, we use the empty corner space to display contextual information and controls. The top left shows the number of similar records being reviewed and the total number of archived records. The color legend is at the bottom right. Controls for interactively editing the peer group within the LikeMeDonuts are at the top right corner.

A set of control buttons are provided for editing the peer group at the record level. At the start, the buttons are disabled. Clicking on ring sectors will select a record subset and enable the "Remove Selected Records" button. As users make edits, the "Undo", "Redo", and "Reset" buttons become available. The removed records are filtered out and excluded in other visualizations immediately.

Ranking Glyph. The role of the Ranking Glyph is to help users understand how similarities and differences for each criterion evolve as they go down the ranked list of similar records. Each glyph represents a criterion and each horizontal bar within a glyph represents a record (Figure 3b). Records are ranked by their similarity to the seed record in all glyphs, with the most similar ones at the top and least similar ones at the bottom. The same consistent color scheme is applied. Bright green bars indicate that the criteria value of those records are identical to the value of the seed record while dark green bars represent records with criteria values within user-specified tolerance ranges. Records with criteria values outside tolerance ranges are shown as gray bars. The glyphs are arranged in the same layout as the criteria controls (Figure 2b) and the record ranked list (Figure 3d). Hovering on a glyph highlights the focused criterion in other visualizations. Records selected in other visualizations will be highlighted in orange in the Ranking Glyph, revealing their positions in the ranked list.

History Heatmap. The History Heatmap summarizes the temporal events of the entire peer group or any selected subset of records. Each row of the timetable represents an event category and each column represents a time period (Figure 3d). In the example of students' academic records, each time period is a semester (e.g., Spring, Summer, and Fall). The darker the color of a cell the more events occurred in the time period, revealing hot spots of activity in black (such as unsurprisingly"Start" in the first semester) and unpopular event in white (e.g., "Advanced Course" in Summer). When users select a subset of the similar records in other visualizations (e.g., by clicking on a ring sector in LikeMeDonuts), their activities will be shown in the history Heatmap using an orange color gradient.

3.4 Reviewing Recommendations of Action Plans

Outcome Estimation. The outcome distribution view (Figure 4b) summarizes the estimated outcomes of the similar records (thick bars) and all archived records (thin bars). It provides an estimation of the most likely outcome of the seed record and the likelihood. Users can select a desired outcome using the radio buttons. They can also compare the seed record's likelihood of achieving the desired outcome against the baseline of all records.

Activity Summary and Action Plan. The activities of the similar records are summarized and shown as part of the timeline of the seed record (Figure 4a). A darker color indicates this activity is more popular in this time period. Users can keep only records having the desired outcome and use their activity patterns as guidance for specifying the action plan. By analyzing the event sequences of those similar records who have achieved the user's desired outcome, EventAction recommends a representative action plan to the user, which is displayed on top of the seed record timeline on the future side (Figure 4c). Users can review the recommended plan and choose to (1) follow the plan without modification, (2) tune the plan to better fit their needs, or (3) use the recommended plan as a reference and design their own plans from scratch. To tune the recommended plan, users can click to add events to the time table or change their numbers of occurrences. When the plan is being changed, EventAction will update the outcome estimation taking the planned events into consideration (Figure 4d). Users can review how the current plan affects the outcome likelihoods in real time.

3.5 Analytical Workflow

EventAction's analytical workflow (Figure 5) was developed and refined based on our observations of user behaviors during empirical studies and case studies. The typical workflow starts from selecting a seed record and the first step is to find a group of similar records. After submitting the similar records, a recommendation model will be computed and users review a recommended action

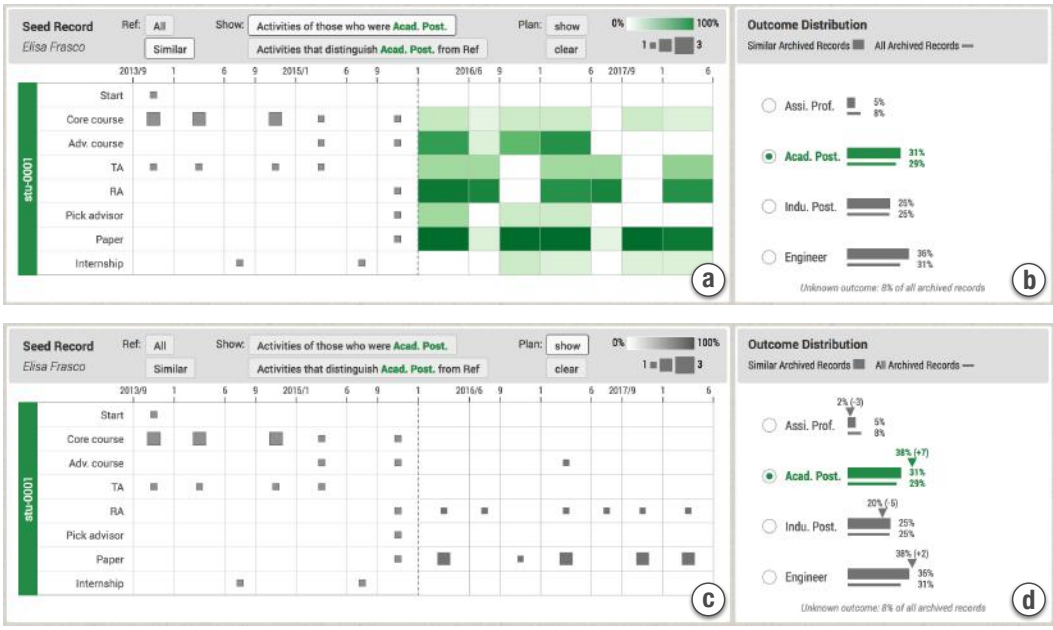

Fig. 4. Components for reviewing and tuning recommendations of action plan: (a) Activity history of the seed record (gray squares on the left) and activities summary of similar records (heatmap on the right); (b) EventAction's estimation of the outcomes of the similar records (thick bars) and all archived records (thin bars) with desired outcome highlighted in green; (c) recommended action plan (gray squares on the right); (d) an updated estimation incorporating the action plan (shown as triangles above the bars). In this example, the recommended plan emphasizes research activities such as RA (research assistantship) and paper. It also suggests taking some advanced courses.

plan. Then, users further refine the plan by directly editing the plan using the activities of similar records as a reference or refining the similar records to generate an updated recommendation.

We also observed many small deviations in the workflow during user studies and case studies. For example, some users changed the order of the steps (e.g., reviewing the recommended plan before refining similar records), some skipped certain steps (e.g., skipped reviewing and refining similar records), and some started refining similar records by keeping only identical records while some started by showing all records. How users perform the analyses depends on many factors such as their familiarity with the interface, the duration of the analysis, and specific datasets and analytical goals. To satisfy different users' needs, EventAction supports flexible analytical workflows. For example, EventAction allows users to skip the step of finding similar records and start by reviewing the recommended plan. In this case, the recommendation will be generated using a set of records retrieved with default similarity criteria.

3.6 Reflections on the Final EventAction Design

The final EventAction system combines ten preliminary interface components for finding similar records [17, 18] and reviewing recommendations of action plans [16]. The overall integration went through a dozen iterations over a six-month period, during which we developed and demonstrated prototypes to our case study partners, gathered their feedback, and discussed improvement plans.

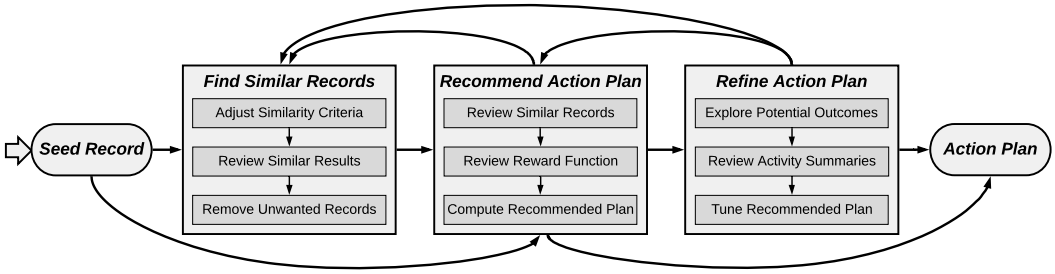

Fig. 5. The analytical workflow of EventAction. The typical workflow starts from selecting a seed record and the first step is to find a group of similar records. After submitting the similar records, a recommendation model will be computed and users can review a recommended action plan. Then, users can further refine the plan by directly editing the plan using the activities of similar records as a reference or refining the similar records to generate an updated recommendation.

To simplify the interface, we added supports (Figure 1c) for expanding the displays progressively so that novice users can start with basic functionalities: the "Basic" configuration only shows the results without any controls, the "Simple" configuration allows users to "Ignore" some criteria using the Similarity Criteria Controls (Figure 2b), and the "Complex" configuration provides all available controls and shows LikeMeDonuts (Figure 3a).

To cope with complex datasets, we added the capability of using a power scale for the sizes of the squares in the Record Timeline (Figure 3d), which improves the visibility of minor categories when the numbers of events in different categories are imbalanced. We also provided options to group the event categories (e.g., interventions, reactions, and outcome as in Figure 8a), so that users can easily differentiate among types of activities and focus on one group at a time when the number of event categories is large.

Finally, we propagated a simplified color scheme to all the visualizations: bright green for exact match, dark green for close match, gray for mismatch, and orange for highlighting. The consistent use of green colors helps users quickly identify similar records across all the interface components. We also added brushing and linking capabilities to coordinate all the visualizations so that users can easily select records of interest from one view and inspect them in others.

4 DESCRIPTION OF THE SYSTEM BACKEND

EventAction's system backend consists of a data pipeline for finding similar records and an automatic algorithm for generating event sequence recommendations. In this section, we describe these two components and report on an experiment evaluating the data pipeline's performance on large testing datasets.

4.1 Data Pipeline for Finding Similar Records

This section describes the data pipeline of EventAction for finding similar records, which needs to execute upon every similarity criteria adjustment and has a great impact on the system's performance and interactive latency [40]. The data pipeline consists of 6 steps, from loading the raw data to showing the results of similar records. The raw data are two tab-delimited text files, one for temporal events and the other for record attributes. Each record (identified by a unique *Record ID*) is represented as a sequence of events. Each event belongs to a particular *Event Category* and is assigned a *Timestamp*. Descriptive information of each record is carried in attributes and stored as a pair of *Attribute Name* and *Attribute Value*. We first describe how EventAction processes

the data through each step of the pipeline. Next, we provide more details about the distance score computation. Finally, we report on an experiment evaluating the performance of the pipeline.

4.1.1 Data Pipeline.

Data Loader. After the analysts load the raw data (identified by Data Name), EventAction creates record instances to organize the event and attribute information of each record and stores them in memory. In each record instance, the event sequence is structured as an Array and the attributes are structured as a HashMap. Record instances are indexed by Record IDs so that they can be retrieved in constant $O(1)$ time. At this step, users need to specify the Seed Record of their analyses.

Time Filter. At this step, users define a time window of the history, for example, from start school until the end of the second school year. EventAction will extract events within this history window from each record and use them for finding similar records. The time filter iterates over the events of all archived records in $O(E)$ time, where E is the total number of events in the dataset.

Criteria Filter. By default, for each similarity criterion marked as "Exact Match" only the archived records that have the exact same value (or exact same pattern for temporal criteria) as the seed record will be retained. If user-adjustable tolerances have been specified, the records' criteria values only need to be within the tolerance ranges to be retained. The tolerance range is represented by a set of values for categorical criteria, the minimum and maximum values for numerical criteria, or a pair of upper and lower bounds of the pattern difference scores for temporal criteria. This step iterates over all archived records and all criteria in $O(R \cdot C)$ time, where R is the total number of records in the dataset and C is the number of criteria of each record.

Ranker. Next, "Close Match" criteria are used to rank the archived records by their similarities to the seed record. A comprehensive distance score is computed for each archived record by first assessing the difference in each criterion and then summarizing them into a single distance score. Both assessing the differences in "Close Match" criteria and computing the summary distance score take $O(R \cdot C)$ time. Ranking the records by similarity takes $O(R \log R)$ using Python's built-in sorting algorithm.

Similarity Filter. Given a Similarity Threshold specified by users, EventAction further removes records that are not similar enough to the seed record (i.e., records with a distance score larger than the threshold). This step iterates over the records and takes $O(R)$ time.

Interface. Finally, the remaining similar records are passed to the visualization views and shown to users. The views also provide interactive controls for users to refine the results.

4.1.2 Distance Score Computation.

For those "Close Match" criteria, a comprehensive distance score is computed for each archived record based on the empirical assumption that the archived records tend to be more different from the seed record if they have (1) nonidentical values for categorical attributes, (2) larger discrepancies in numerical attribute values, and (3) larger deviations in activity patterns. The algorithm first assesses the differences for each criterion and then summarizes them into a single distance score.

Categorical criteria: For each categorical criterion $cc \in C$, we define the difference score between an archived record r and the seed record s as:

$$\Delta_C(cc, r, s) = \begin{cases} 0 & v(cc, r) \in t(cc, s) \\ \alpha & v(cc, r) \notin t(cc, s) \end{cases}$$

where $v(cc, r)$ returns the value of the categorical criterion (cc) for a given record and $t(cc, s)$ returns the set of categorical values in the user-specified tolerance range (or $\{v(cc, s)\}$ if the tolerance is not specified). The difference score is set to α when the value of the given record (r) is not within the tolerance range. We let $\alpha = 0.5$ to keep a balance between categorical and numerical criteria, but the optimal value depends on the data and analysis.

Numerical criteria: For each numerical criterion $nc \in N$, the difference between an archived record r and the seed record s is formulated as:

$$\Delta_N(nc, r, s) = \begin{cases} |v(nc, r) - t_u(nc, s)| & v(nc, r) > t_u(nc, s) \\ |v(nc, r) - t_l(nc, s)| & v(nc, r) < t_l(nc, s) \\ 0 & otherwise \end{cases}$$

where $v(nc, r)$ returns the value of the numerical criterion (nc) for a given record and $t_u(nc, s)$ and $t_l(nc, s)$ returns the upper and lower bound of the user-specified tolerance range, respectively. When the tolerance of nc is not specified, we have $t_u(nc, s) = t_l(nc, s) = v(nc, s)$. Before the computation, values of each numerical criterion are standardized by scaling to range $[0, 1]$.

Temporal criteria: For each temporal criterion $tc \in T$, we compute a pattern difference score $v(tc, r)$ for each archived record r, reflecting its difference from the seed record s in activity patterns:

$$v(tc, r) = \|\mathbf{p}(tc, r) - \mathbf{p}(tc, s)\|$$

where $\mathbf{p}(tc, r)$ returns a two-dimensional vector (x=time, y=event category) representing the activity pattern of r. The values represented in $\mathbf{p}(tc, r)$ are the numbers of occurrences of each event category during each time period. The minus operation computes the Euclidean distance between two vectors and the output of $v(tc, r)$ is a numerical value. This allows us to reuse the difference function for numerical criteria and let $\Delta_T = \Delta_N$.

Finally, we summarize a comprehensive distance score for each pair of archived record r and the seed record s based on weighted Euclidean distance [14]:

$$distance(r, s) = \sqrt{\sum_{cc \in C} w_{cc} \Delta_C^2(cc, r, s) + \sum_{nc \in N} w_{nc} \Delta_N^2(nc, r, s) + \sum_{tc \in T} w_{tc} \Delta_T^2(tc, r, s)}$$

where $w \in [0, +\infty)$ is the weight assigned to a criterion.

4.1.3 Time Complexity Analysis.

Finding similar records is a task that frequently repeats during analyses. We have conducted experiments to evaluate its performance. In theory, the overall time complexity of the EventAction data pipeline is $O(E + R \cdot (C + \log R))$, where E is the total number of events, R is the total number of records, and C is the number of criteria of each record. To provide a sense of timing, we conducted an experiment using synthetic datasets of varying numbers of records (100, 200, 400, 800, 1,600, 3,200, 6,400, 12,800, 25,600, and 51,200) and numbers of criteria (10, 20, and 30). In each dataset, half of the criteria were categorical and the other half were numerical. All criteria were set to "Close Match." On average each record contained a sequence of 40 events and thus the total numbers of events in the testing datasets are 4,000, 8,000, 16,000, 32,000, 64,000, 128,000, 256,000, 512,000, 1,024,000, and 2,048,000.

Figure 6 reports the average runtime of 100 repetitions tested on each dataset. All tests were performed on a machine with a 2.3 GHz Intel Core i7 processor with 16 GB 1600 MHz DDR3 memory. The results show that the time for finding similar records grows almost linearly as the number of records (R) increases by a factor of two, and the growth rate was mainly determined by the number of criteria (C).

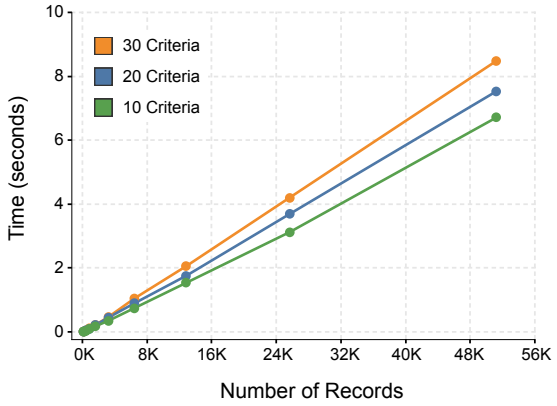

Fig. 6. The average runtime of the EventAction data pipeline on synthetic datasets of varying numbers of records and numbers of criteria. The results show that the time for finding similar records grows almost linearly as the number of records increases by a factor of two, and the growth rate was mainly determined by the number of criteria.

4.2 Automatic Recommendation of Event Sequences

In this section, we introduce an extension to EventAction for generating recommendations of action plans automatically. After a user has selected the similar records, the event sequences of those similar records who have achieved the user's desired outcome will be analyzed and a representative action plan will be recommended to the user. Users can review the recommended plan and choose to (1) follow the plan without modification, (2) tune the plan to better fit their needs, or (3) use the recommended plan as a reference and design their own plans from scratch. In this section, we describe the algorithm for generating sequence recommendations and the challenges and solutions for integrating it into EventAction.

4.2.1 Sequence Recommendation Algorithm.

Our sequence recommendation algorithm was based on Markov decision processes (MDP) and used an implementation provided by Theocharous et al. [74, 75]. This section summarizes the basic process of generating recommendations of event sequences (illustrated in Figure 7). Full algorithmic and evaluation details can be found in the original papers. MDP-based models are widely used in applications as a mathematical framework for solving sequential decision problems (e.g., navigating a robot) [80]. While recurrent neural networks (RNN) are emerging in sequence modeling applications, we chose MDP for two main reasons. First, the datasets of our current use cases are relatively small (e.g., thousands of student records or patient events) and the set of similar records for model training are even smaller (hundreds of records). While most recurrent neural network models require a large dataset to train (typically around 100k records) [72], MDP-based models show good performance on small datasets in terms of accuracy and model building time [65]. Second, our recommendation model needs to be retrained every time the set of similar records have been updated. In our use cases, MDP-based models can generate recommendations within a reasonable latency (typically less than one minute). Nevertheless, we believe deep learning models have great potentials in future EventAction applications which are likely to require large-scale datasets and powerful machines.

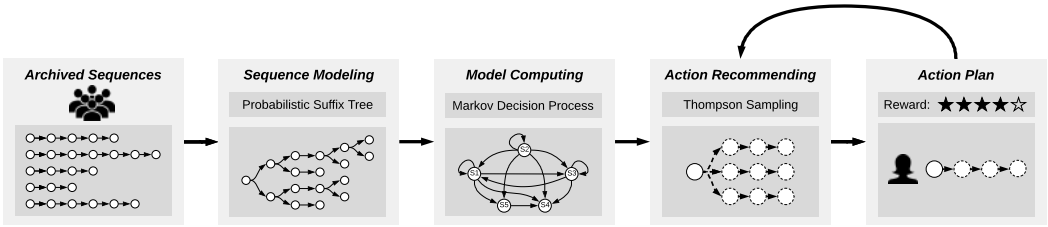

Fig. 7. An illustration of the sequence recommendation algorithm.

Sequence Modeling. The first step is to model archived event sequences using a probabilistic suffix tree (PST), which takes into account a record's activities so far to recommend the next action. PST provides a compact way of modeling temporal patterns that compresses the input sequences to accelerate computation. Each node in a PST encodes a frequent suffix of history events and is associated with a probability distribution of the next events. Given a PST model and the history event suffix $S = (e_1, e_2 ... e_t)$, the probability of the next event can be estimated as $P(e_{t+1}|S)$. Our implementation used the *pstree* algorithm [21] in R language.

Model Computing. After building the PST, the next step is to create MDP models. The MDP model can be computed directly from the PST, where the states of the MDP are nodes of the PST and the state transition probability is derived from the longest paths in the PST. Specifically, given a history event suffix $S = (e_1, e_2 ... e_t)$ available as a node in the PST tree, the model computes the transitioning probability from each node to every other node by identifying the longest suffixes in the tree for every additional event that an action can produce.

Action Recommending. The last step is to find the optimal policies generated by the MDP models for generating recommended event sequences. Our implementation uses Thompson sampling [76], which is a heuristic approach for choosing actions that address the exploration-exploitation dilemma in the multi-armed bandit problem [6]. In particular, Thompson sampling is capable of choosing the next actions in real time to maximize the "expected reward" as specified on each state (usually provided in the dataset or specified by users). Gopalan and Mannor [23] have extended Thompson sampling to be applicable to MDPs. Specifically, in each round of sampling, an action a^* is simulated according to the probability that it maximizes the expected reward $E(r|S, a^*)$, where $S = (e_1, e_2 ... e_t)$ is the suffix of history events. Theocharous et al. [74] conducted experiments to compare Thompson sampling against a greedy planning strategy and found that Thompson sampling runs faster and can produce more rewards than the greedy approach.

4.2.2 *Integration into EventAction.*

We describe the three major challenges and our solutions for integrating the automatic recommendation algorithm into EventAction.

Event Co-Occurrence. The sequence recommendation algorithm was originally designed for recommending travel plans, where each event represents a place to visit without any overlapping [74]. However, event co-occurrences commonly exist in many other application domains where multiple events occur or being logged at the same time. For example, a patient may take multiple drugs together and a student may attend multiple classes during a day. Due to the use of probabilistic suffix tree, the original sequence recommendation algorithm was not capable of modeling or recommending sequences with co-occurred events.

Our implementation overcomes this challenge by transforming the co-occurred events into event episodes. Each episode is an unordered combination of events with possible repetitions, represented by a vector $E = (|e_1|, |e_2| \ldots |e_n|)$. Event episodes are categorized by its event compositions and the raw event sequences are encoded into sequences of event episodes, which can be used by the sequence recommendation algorithm. The recommended plan also consists of event episodes and is decoded back to the original events before presenting to users.

Reward Function. In the use case of recommending travel plans, the "reward" for visiting each place (i.e., event) can be assessed based on its ratings from past visitors, which can be easily obtained from online services (e.g., TripAdvisor or Google Maps). However, subjective ratings for events are generally not available and difficult to collect in many other domains. To make the sequence recommendation algorithm usable even when rewards are not provided in the dataset, we defined a default reward function by counting the popularities of the events of records that are similar to the seed record and have achieved the desired outcome. This reward function makes the assumption that the event popularities are correlated with the outcomes. Users are encouraged to verify this assumption or define their own reward functions.

Scalability. The time complexity for computing the Markov Decision Process mainly depends on the number of nodes in the probabilistic suffix tree, which grows exponentially as the number of unique sequences increases. A detailed performance evaluation has been conducted by Theocharous et al. [74]. To reduce the latency, users can choose to classify the event episodes and only keep N representatives. A larger N will produce more tailored recommendations but cost longer computation time. The default value of N is 20 and the computation typically takes less than one minute. However, the optimal setting depends on specific datasets and analytical goals. In addition, the recommendation algorithm is run in a separate process in parallel with the system's main process, so that users can keep exploring during the computation. After the computation completes, a recommended plan will be displayed on top of the seed record timeline on the future side (Figure 4b).

Limitations. We did not evaluate the accuracy of the sequence recommendation algorithm in our application scenarios due to the lack of ground-truth for "the most appropriate set of similar records" or "the most appropriate reward of each event category." Thus, it is difficult to assess the accuracy of the recommendation algorithm in terms of "what is the best plan." When designing EventAction, we focused less on achieving the best possible algorithmic accuracy but more on giving users controls and allowing users to develop personalized plans that they are more likely to follow. In most cases, the recommended plans serve as reasonable starting points as they are derived from the data and with theoretical supports.

5 CASE STUDIES

The case study which was our motivating case study (i.e., the education case study with a student advising scenario), as well as feedback based on early versions of EventAction has already been reported in early papers [16, 18]. This section reports on three new case studies that illustrate the potential use of EventAction in healthcare and marketing.

Following the Multi-dimensional In-depth Long-term Case studies (MILCs) procedure [67] all our case studies were conducted with real users and using real-world datasets. They provided evidence of the value of generating event sequence recommendations based on personal histories and helped us produce design guidelines for the construction of event sequence recommendation user interfaces and usage guidelines for mitigating the ethical issues in dealing with personal

histories. At the end of this section, we also describe three incomplete case studies which may help potential users identify conditions for suitable applications of EventAction.

5.1 Medical Intervention Planning for Health Coaches

This case study was conducted with two health analysts using real-world patient health records. This two-month case study included biweekly discussions, interviews, data preparation, system deployment, and data exploration. We provided training and necessary guidance, and answered questions over the meetings and interviews. Our goal was to investigate how EventAction might help health coaches choose personalized health interventions.

Task. This case study was conducted with health analysts working with a population management company. The company hires health coaches to monitor patient health conditions with sensors. When an alert shows up, the coach needs to help the patient resolve it. The study goal was to evaluate if EventAction might help determine the best way to resolve those health alerts.

Health coaching traditionally encompasses five principal roles: (1) providing self-management support, (2) bridging the gap between clinician and patient, (3) helping patients navigate the health care system, (4) offering emotional support, and (5) providing continuity of care. While health coaches have always had to interpret information and decide on engagement strategies, with the introduction of mHealth tools, an effective health coach must be able to interpret more frequent, voluminous and diverse data, in effect becoming a data analyst, in addition to a behavior change agent. Health coaches must decide: who needs attention, the priority of outreaches, what mode of contact may work best, and what approach may be appropriate. Traditionally, this was accomplished with judgment and limited data, but innovative analytics incorporating pervasive data and individual differences (e.g., demographics) allow health coaches to make these decisions based on what worked for similar cases, offering new-found possibilities for precision healthcare through mHealth.

Data. The research setting includes 107 health insurance plan members that were enrolled in a mHealth care management program. These plan members are age 34-66, with poorly controlled chronic disease, principally congestive heart failure, as identified by the plan using healthcare claims data. The cohort consists wholly of Medicaid managed care plan members. It can be argued this population faces special challenges with the social determinants of health, factors such as housing, transportation, access to food, safe neighborhoods. While the results need to be considered in light of these differences from affluent populations, the treatment activities chronic disease patients should adhere to and the role of health coaches are similar. The data used in this case study included demographics (gender, age, weight), test results (diastolic, SpO2, and systolic blood pressure), outreaches (1,004 events including coaching call, voice message, text message, and others), and care gaps (2,626 alert events).

Analysis. During the case study, the health analysts selected a current patient (46 years old, male) as the seed record. EventAction retrieved and displayed the profile and recent activities of the patient. The analysts immediately noticed that while the weight of the patient was in the normal range, he had extremely high diastolic and SpO2 readings. They adjusted the weights of these two similarity criteria to find patients with similar test results.

From the timeline, the analysts found the patient had health alerts every day during the last three days, indicating that health coaches' attention was needed for resolving the alerts. However, as clearly shown in the timeline, the health coach only called the patient once on the third day, which was delayed and unexpected. The analysts created a new similarity criterion to reflect this pattern of not being contacted during the first two days of alerts. The top 20% most similar records

were selected as the peer group. The outcome distribution view showed that the health alerts of 69% of those similar patients got resolved within 5 days, which was slightly above the baseline of 61%.

To develop a health intervention plan for resolving the alerts of the patient, the analysts reviewed the activity summary of those similar patients. The heatmap showed that 69% of the similar patients will continue to have alerts on the fourth day and the number stays above 50% until the eighth day. Furthermore, the most common health interventions for those patients were a daily coaching call. The analysts switched to show activities distinguishing patients who had their alerts resolved within 5 days from others. Green hotspots showed up for coaching call during the fifth, sixth, and seventh days, indicating that interventions were most effective during these periods. The analysts specified a plan using these insights and the estimated likelihood of resolving the alerts within 5 days increased by 8%.

Feedback. Reviewing EventAction with health analysts provided actionable insights. A powerful component of EventAction is that it allows for hypothesis testing of patient results, as one health analyst said: *"A health coach or care manager can pose questions to the data, such as what could happen when similar patients to the patient under inquiry did X?"* The other analyst appreciated the rich similarity criteria controls and commented that *"the system allows for addressing population health strategies through easily identifying and segmenting patient cohorts by customizable data parameters."*

The study indicated that a more specific interpretation of results was needed. We envision that this would be built into training materials and that tooltips could be included to display advice when the pointer hovers over it. The clear flagging of strategies that are recommended or not recommended were additional features highlighted for development. Besides, the use of a tool that embeds peer comparisons for health naturally raised privacy concerns that one may be exposing peers unnecessarily. Additional work to conceive appropriate anonymization for large-scale implementation is needed. Finally, the heatmap meaning was not clear at first and was revised. Darker colors made it easier to see where similar patients achieved desired results.

We recognize the limitation that the results are for one mHealth care management system, limited data, and involve feedback from a small number of health analysts; however, this exploratory case study is the first attempt at exploring the use of such systems.

5.2 Campaign Planning for Marketing Analysts

This section reports on two case studies conducted with 5 marketing analysts and using real-world event sequence datasets. Two of the analysts focused on email campaigns, two on cross-channel marketing, and one on web analytics. Each case study lasted about a month consisting of interviews, data preparation, system deployment, and data exploration. During the case studies, we provided training and necessary guidance, and answered questions. The study goal was to investigate how EventAction can help marketers prescribe personalized marketing interventions. Figure 8 illustrates a synthetic dataset of customer records. Since marketing datasets usually contain large numbers of records, it is impossible to precisely make plans for each customer. The marketing analysts in the case studies evaluated EventAction by selecting seed record that is representative of a type of customers so that the action plan will be applicable to them as well.

5.2.1 Customer Onboarding.

After customers start the product trial, the marketers will send them a series of 5 onboarding emails to help them learn to use the product and to engage them to make purchases after the trial period. Each of the 5 emails provides different content, including welcome notes, product

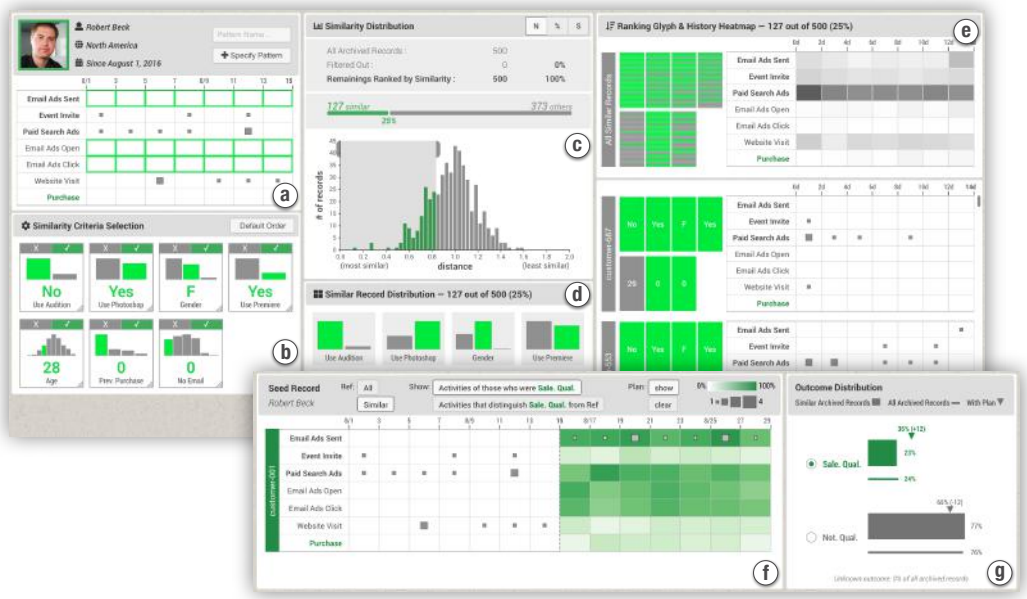

Fig. 8. This figure shows a synthetic dataset illustrating the "Channel Attribution Analysis" case study. For privacy constraints, the real datasets used in the case studies are not shown in the figure. A seed record and 500 archived customer records are displayed. Marketing activities are related to sending email ads, event invitation, and paid search ads (a). Record attributes include the customers' genders, ages, and previous product purchases (b). Two types of outcomes are defined: "Sales Qualified" and "Not Sales Qualified" (g). All record attributes are used as similarity criteria by default and a new criterion is created to capture the temporal pattern of having no email-related activities (a). The top 25% most similar records are selected as the peer group (c). An action plan of sending the customer more email ads is specified (f) and the likelihood of becoming "Sales Qualified" increases by 12% (g).

promotions, tutorials, and learning resources. In this case study, the analysts wanted to make plans for sending onboarding emails to new customers so as to increase their engagement.

Data. The analysts provided a dataset of 25,000 archived records of past customers who have received the 5 onboarding emails. The dataset contains about 112,000 events tracking the send, open, and click of each email. We used a sample of 500 records and 8,191 events in the case study. Only one record attribute existed in the dataset indicating the regions of the customers. The outcome was defined by the number of emails that customers clicked any links in, such as links to the product purchase website or to tutorial videos, which is an indicator of their engagement during the product trial. The outcome was categorized into "0 click", "1-2 clicks", and "3-5 clicks", where "3-5 clicks" was the most desired one.

Analysis. The analysts selected a seed record for a customer who had received and opened the first two emails but did not click on any links. They wanted to make a plan for the subsequent emails that may lead to the outcome of "3-5 clicks." They started by specifying a "no click" pattern and only keeping customers having this pattern. Then, they selected the top 30% most similar records as the peer group and continued to review guidance for planning.

The analysts opened the activity summary view to review the email sending patterns of all archived records. The heatmap showed hotspots approximately every 7 days with some variations, which was expected by the analysts. From the outcome distribution view, the analysts realized that the seed record's likelihood of clicking 3-5 emails was only about 3%, which was much worse than the baseline of all archived records. The analysts decided to lower their expectations and changed the desired outcome to "1-2 clicks."

Then, they reviewed activities that distinguish customers who had "1-2 clicks" from others in the peer group. A green hotspot for email #3 showed up three days after sending email #2. About 11% more similar customers who received email #3 on that day will make 1-2 clicks during the onboarding. If they also open that email, the difference will further increase to 14%. The analysts checked the content of email #3 and found that it was featuring learning resources and tutorials for the product. They explained: *"We thought it might be an important email and now EventAction provides evidence for it."* Following these findings, the analysts specified a plan for sending the subsequent emails. EventAction estimated an 11% increase in the seed record's likelihood of making 1-2 clicks.

5.2.2 Channel Attribution Analysis.

In this case study, the marketing analysts wanted to understand which campaign channels will be the most effective for converting a current customer into sales qualified, which means they are ready for the sales team to reach out.

Data. The analysts prepared a dataset of 997 customer records and 26,472 events. The record attributes included which product was promoted and the region of the campaign. Campaign activities included "event invitation", "paid search ads", and "email ads sent". Customers' activities included "email ads open", "email ads click", and "website visit". The outcome was defined by whether or not a customer became sales qualified judged by the sales team.

Analysis. The analysts select a seed record who actively opened emails but never visited any product websites during the past 5 months. They reviewed the profile of the customer and found that their past interactions with this customer were mainly by email with only a few "event invitations" and no "paid search ads." They created a new similarity criterion to reflect this pattern and selected the top 20% most similar records as the peer group.

The analysts immediately noticed that in the following 5 months those similar customers usually continue to actively receive and open emails. Their likelihood of becoming sales qualified was slightly below the baseline but still promising. The analysts switched to show activities distinguishing those who became sales qualified from others. Green hotspots showed up in the 6th and 7th months for "event invitation", "email ads sent", and "email ads click" indicating that sending out event invitations and campaign emails soon may help improving the outcome. The analysts specified a plan using these insights and the estimated likelihood increased by 10% which outperformed the baseline.

5.2.3 Feedback.

Pseudo A/B Testing. In both case study, the marketing analysts found EventAction useful for testing hypotheses based on historical data. For example, one commented that *"EventAction allowed me to simulate plans and get results immediately. This can help me select variables for A/B testings."*

Temporal Information. All marketing analysts liked EventAction's visual and interactive way for exploring the temporal information as one said *"I can see the data directly."* The analysts of the channel attribution study also applaud that EventAction introduced a new time dimension for their

attribution analysis as one explained *"it not only informed us about which channels were important but also showed how the importance evolves over time."* In addition, *"EventAction enabled us to filter the records using temporal patterns, which helps getting more precise results"* another analyst added.

Automatic Planning. The analysts were excited about EventAction's automatic plan recommendation feature because *"it will save a lot of time and effort in the long term."* However, they prefer to learn more about the mechanism before relying on it in real tasks. They suggested a workflow of showing the recommended plan at the beginning and allowing users to modify it during the analysis, which is a workflow deviation supported by EventAction (Section 5).

5.2.4 Challenges and Solutions.

Through the process of the two case studies, the analysts have highlighted the challenges in analyzing customer records and planning marketing interventions. These challenges lie in both the uniqueness of customer records and specific marketing tasks. We cover the 4 major challenges and discuss our solutions.

Limited Record Attributes. Unlike patient or student records, customer records are usually anonymous without details such as demographics, diagnoses, or surveys. The available record attributes are usually very limited which makes it difficult to profile the customers and design personalized campaign strategies. EventAction addresses this challenge by using customer's activity patterns to identify similar customers and guide the planning. For example, given a customer who opens campaign emails but never visits the product website, marketers can find similar customers having this activity pattern and explore what campaign strategies worked the best for them.

Visualizing Complex Temporal Data. Temporal data in the marketing domain are difficult to visualize due to their complexities in three aspects: (1) the number of event categories is large capturing various campaign-related activities; (2) the numbers of events in categories are very different, ranging from hundreds of emails sent to only one or two purchases; (3) many events occur at the roughly same time causing severe overlaps and visual clutter.

EventAction's timeline view can effectively handle event co-occurrences (3) by aggregating events in each time period. However, since it uses the sizes of the squares to show the numbers of events, popular categories will dominate the view (2), making squares in minor categories invisible. We addressed this issue by using a power scale $size = sqrt(num)$ when the range of the sizes is large. We also grouped the event categories into three groups to help users focus on one group at a time (1): interventions, reactions, and outcome. However, a more scalable timeline design is still needed to fully address aspect (1) when the number of event categories is large within a group.

Large Number of Records. A marketing dataset may contain millions of customer records, which can significantly slow down the computation and rendering. EventAction mitigated this issue by only visualizing similar records. To accelerate the similarity computation, future work could be conducted to investigate other techniques such as clustering and comparing records in groups.

Slow and Expensive A/B Testing. Conducting A/B tests to examine different campaign strategies may cost significant resources and take a long time when the number of variables is large. EventAction provides a low-cost approach allowing marketers to quickly simulate different plans using historical data and get immediate results. The actual A/B testing will only need to cover strategies with promising results or low confidences (e.g., very few archived records matched the criteria).

6 DISCUSSION

This section describes the design and usage guidelines produced through our studies. Then, it discusses promising directions to extend our current software prototypes and studies.

6.1 Guidelines

We describe five design guidelines for the construction of event sequence recommendation user interfaces and three usage guidelines for mitigating the ethical issues in dealing with personal histories. These guidelines are produced through our empirical studies of interface components and case studies in three domains, including education, marketing, and healthcare.

6.1.1 Design Guidelines.

G1. Center the interface design on the seed record. Unlike many other event sequence visualization tools [24, 50, 82, 83], the analytical workflow of EventAction is oriented by a seed record. Centering the interface design on the seed record emphasizes the workflow and keeps users focused on the tasks of finding similar records and making action plans for the seed record. For example, when designing the LikeMeDonuts, we placed an image of the seed record at the center, which provides a visual reminder that all the information is relative to that person. The thickness of each donut ring and the color of each cell are meaningful in achieving the goal of finding similarity or differences. Users found this design clearly illustrated the purpose of the interface and they tended to move important criteria closer to the image to be focused.

G2. Increase algorithm transparency with visualizations and user controls. Our study results showed that increasing the algorithm transparency of sequence recommender systems can increase users confidence and engagement, even at the cost of added complexity. How people perceive the similarity between personal records is subjective, depending on their preferences, experiences, and beliefs, leading some observers to dismiss this as a slippery notion [15]. It is possible to define a set of initial similarity criteria but users should be able to review and adjust those criteria for specific applications. For example, EventAction provides visualizations to help users review similar records and provides controls for users to adjust similarity criteria. This paper focuses on making critical life decisions in which users demand more controls and context even at the cost of added complexity [17, 29]. Our designs and findings require more effort than is typical in entertainment and shopping recommender systems, which are used for less critical decisions. While our work focused primarily on the design of interfaces to help users review and tune the peer group and recommendation, there may be better algorithms to generate the initial recommendation, for example, using recurrent neural networks [72].

G3. Show both individual-level details and group-level overviews. Reviewing and refining the results of similar records are key steps in the analytical workflow of event sequence recommendation. The interface should provide both individual-level details and group-level overviews so that users can efficiently review and refine similar records using both record attributes and temporal events. In addition, the group-level overviews should allow users to track and review a group of records that share similar values across multiple criteria, so that users can estimate the group size, explore how those records are distributed in other criteria, and refine the results by removing the group when necessary. For example, EventAction uses a ranked list to show individual details and provides three visualization components for reviewing and refining peer groups, including LikeMeDonuts, History Heatmap, and Ranking Glyph.

G4. Include both record attributes and temporal activities. Electronic records of personal histories (e.g., patients, students, historical figures, criminals, customers, etc.) consist of multivariate

record attributes (e.g., demographic information) and temporal activities (time-stamped events such as first diagnosis, hospital stays, interventions). To compare personal records and define similarity criteria, it is important to take into consideration both record attributes and temporal activities. In particular, we found temporal activities play a more fundamental role in some application domains such as digital marketing, where the records are usually anonymous without detailed attributes such as demographics, diagnoses, or surveys. In EventAction, both record attributes and temporal activities are used as features to identify similar records and provide appropriate recommendations. It allows users to select records that have similar attributes and start with a similar event sequence, and then see how different choices of actions and the orders and times between them might lead to users' desired outcomes.

G5. Support flexible analytical workflows to satisfy different users' needs. We noticed many different workflows in our user studies, deviating from the typical analytical workflow (see Section 3.5). To satisfy different users' needs, the interface should support flexible analytical workflows. For example, EventAction allows users to skip the step of finding similar records and start by reviewing the recommended plan. In this case, the recommendation will be generated using a set of records retrieved with default similarity criteria.

6.1.2 Ethical Issues and Usage Guidelines.

Reviewing ethical issues is important in dealing with personal histories. We discuss the ethical issues we encountered in our studies and describe three usage guidelines for mitigating those issues. EventAction provides a startup screen that reminds users of potential issues and biases in the data (Figure 9).

G6. Use rich, large, and representative data. The holy grail of recommender systems is to convert recommendations into users' actions. Providing reliable recommendations has the potential to increase users' trust in the system and thus motivate actions. The reliability depends on the

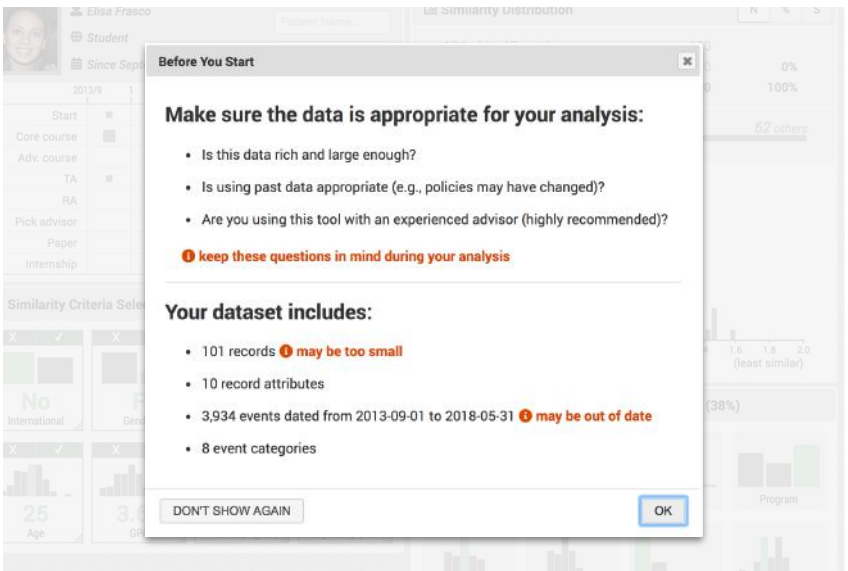

Fig. 9. EventAction's startup screen that prompts usage guidelines and identifies potential issues and biases.

quantity and quality of the data available. To better profile the seed record and find accurate similar archived records, the data describing each record must be rich; and to find sufficient similar archived records, the data volume must be large and representative. Biases may be introduced when the available data do not represent people adequately and there are few similar records exist, or when there are errors or missing attributes in the data [54]. In those cases, it is important to ensure that the algorithm's confidence in generating the recommendation and the user's confidence in following the recommendation remains appropriately low.

G7. Remind users that it is okay to be unique among past paths. Overconfidence can also be an issue. While most students, patients, and others who must make life choices may be eager to follow the paths of predecessors, there are dangers to such an approach. Decision-makers who consult databases of predecessors risk repeating old paths which are no longer relevant because past histories of bias have been rectified or because circumstances have changed. While there may still be lessons from the past, users need to be reminded that their history is unique and that breaking from past paths may be a powerful way to distinguish themselves.

G8. Encourage collaborative use with an experienced advisor. Bad data that reinforces existing biases may be taken as truth and data that challenges them dismissed. Will a poorly performing student be discouraged when seeing the outcome of similar students? Or will a high achieving "anomalous" student in a poor achievement cohort set her horizon too low? Those issues argue strongly for collaborative use where the advisee is working alongside an experienced advisor who can interpret the results or judge data quality. However, advisors' guidance will not solve all problems since they are also vulnerable to biases [7]. EventAction mitigates this issue by giving transparent data access to both advisors and advisees and involving them in the decision-making process.

6.2 Incomplete Case Studies and Limitations

EventAction was used in three other case studies that were not completed for a variety of limitations. This section describes those incomplete case studies to help potential users identify conditions for suitable applications of EventAction.

Too Sparse Temporal Events. A transportation case study used a dataset of emergency responders' activities during auto accidents. The study partner wanted to use EventAction to develop rescue plans for ongoing emergencies by finding similar previous accidents. The temporal events in the dataset consisted of hundreds of categories, which were hand typed by operators and included detailed information such as the names of the responders. EventAction was able to load and visualize the dataset. However, since the events were categorized into too many categories, each category only contained one or two events in several time periods, making it difficult to find valid common patterns or generate reliable recommendations. The study partner was encouraged to find strategies to aggregate the event categories, but decided to change how the data was recorded (e.g., asking the operators to select from a list of possible event categories instead of typing). Not enough data has been accumulated following the new procedure yet.

Too Complex Temporal Pattern Search. One healthcare case study was incomplete due to the complexity of the needed temporal pattern specification. This case study used a dataset of patients' electronic health records. Each record consisted of a patient's complete medical history for years and contained thousands of detailed events such as hospital visits, prescriptions, and health examinations. The study partner wanted to evaluate if EventAction could find similar patients and help doctors prescribe treatments. EventAction was first used to explore a small sample with around 50 events in each record and was able to find reasonably similar records. However, after including

all the events, each patient's temporal activities, spanning many years, became too complex and unique, making it difficult to identify common patterns between the similar records and the seed record using the current EventAction interface. The study partner concluded that they needed to first simplify the dataset (such as extracting events with a time window and coalescing hidden complex events into one [19].) They also decided to develop their own custom similarity search for the specific pattern search they needed [37].

No Suitable Outcome Data. Another healthcare case study was incomplete because the study partners were unable to find suitable outcome data. The case study was conducted with three health analysts using a dataset of medical activities recorded in the emergency room. The analysts wanted to evaluate if EventAction could be used to recommend possible treatment plans for a current patient by finding similar previous patients. After a few visits and meetings, we were able to build an initial EventAction demo to illustrate the process of finding similar patients. However, the analysts then realized that they had not collected data about outcome (e.g., survived or died). Since EventAction requires a clearly defined outcome attribute in each record to generate recommendations, the analysts decided to pause the case study in order to gather the outcomes.

6.3 Future Directions

Our goal for this research was to explore the research possibilities for explainable event sequence recommendations and open up new directions for future researchers. While our case studies were promising, EventAction can be further enhanced:

Scaling Up. Scalability becomes an issue for most interactive visualizations as the size of the data grows. While using powerful machines can accelerate the computation and rendering, reducing human effort in analyzing larger datasets remains challenging for EventAction. We propose three future directions to support analyses of extremely large datasets, such as millions of online customer records. First, while making action plans for one advisee at a time is the typical scenario in many application domains (e.g., healthcare and education), users from several other domains, such as digital marketing, requested support for handling a seed group (i.e., a group of records of interest). Then, marketers can explore archived customers similar to the seed group and develop campaign strategies to increase the purchase rate of the group. Second, similarity searches often return too many matched and partially matched records. Although EventAction presents the results of similar records in a ranked list with the most similar ones on the top, it still costs users extra time to explore and find useful information. To resolve this "information overload" problem, database research has been done to cluster or categorize query results into meaningful groups. Third, when the number of criteria grows larger, showing all criteria at once is likely to overwhelm most users. Automatically selecting two or three criteria to start may be useful [47, 84].

Supporting Collaboration. Unlike existing collaborative visualization systems as summarized by Isenberg et al. [31], one unique challenge in supporting collaboration in EventAction is that the collaborators play asymmetric roles: (1) advisors are usually familiar with the system and thus can fully understand the visualizations and confidently use the controls while advisees are typically novice users who prefer to start with a simple interface, (2) advisors are privileged to review archived records with private information while advisees should only see de-identified data or aggregated summaries, and (3) advisors have knowledge about domain policies and previous professional experience while advisees know better about their own personal preferences and needs. Developing an asymmetric collaboration framework will likely increase advisees' engagement in using EventAction and also benefit similar software tools for student advising, patient caring, and client consulting.

Celebrating Diversity. Components of EventAction could also be put to use for other applications. For example instead of tuning the peer group to be as similar to the seed record as possible, it could be tuned to be diverse. Diversity can drive innovation in teams [30]. An organization may need to assemble a panel of peers to review the grievance brought up by an employee. In this case, the group of peers needs to be close to the employee but diverse enough to include members from multiple divisions of the company, genders, backgrounds, and with some age and background variations. One solution is to extend EventAction's search algorithm to include both "similarity criteria" and "diversity criteria." Then, clusters can be detected in the search results and representative records can be selected from each cluster to achieve diversity.

7 CONCLUSION

Our contributions grow out of our experience in the design, implementation, and evaluation of EventAction, an interactive prescriptive analytics system with a systematic analytical workflow, to assist users in making action plans that elicit high user confidence. Empirical studies in two domains have provided evidence of the effectiveness of generating event sequence recommendations based on personal histories. Through the design, implementation, and evaluation of EventAction, we have produced design guidelines for the construction of event sequence recommendation user interfaces and usage guidelines for mitigating the ethical issues in dealing with personal histories. We believe this paper opens new avenues of research in explainable event sequence recommendations based on personal histories that enable people to make better decisions for critical life choices with higher confidence.

ACKNOWLEDGMENTS

We thank all the study participants for their valuable feedback, in particular, Sana Malik, Eunyee Koh, Georgios Theocharous, Jason Okui, Seth Powsner, Jeff Belden, Evan Golub, Jennifer Story, Matt Scharf, Peggy Hu, Manuel Stevos, Ryan Hobson, and Jonathan Bush. We appreciate the partial support for this research from Adobe Research.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering*. 3–14.

[2] F Gregory Ashby and Daniel M Ennis. 2007. Similarity measures. *Scholarpedia* 2, 12 (2007), 4116.

[3] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential pattern mining using a bitmap representation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 429–435.

[4] Ragnar Bade, Stefan Schlechtweg, and Silvia Miksch. 2004. Connecting time-oriented data and information to a coherent interactive visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 105–112.

[5] Eftychia Baikousi, Georgios Rogkakos, and Panos Vassiliadis. 2011. Similarity measures for multidimensional data. In *IEEE International Conference on Data Engineering*. 171–182.

[6] Oded Berger-Tal, Jonathan Nathan, Ehud Meron, and David Saltz. 2014. The exploration-exploitation dilemma: A multidisciplinary framework. *PLoS ONE* 9, 4 (2014), e95693.

[7] Brian H Bornstein and A Christine Emler. 2001. Rationality in medical decision making: A review of the literature on doctors' decision-making biases. *Journal of Evaluation in Clinical Practice* 7, 2 (2001), 97–107.

[8] Derek Bridge, Mehmet H Göker, Lorraine McGinty, and Barry Smyth. 2005. Case-based recommender systems. *The Knowledge Engineering Review* 20, 3 (2005), 315–320.

[9] Paolo Buono, Catherine Plaisant, Adalberto Simeone, Azizah Aris, Ben Shneiderman, Galit Shmueli, and Wolfgang Jank. 2007. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *International Conference on Information Visualization*. 191–196.

[10] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12, 4 (2002), 331–370.

[11] William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the Annual Symposium on Document Analysis and Information Retrieval*. 161–175.
[12] Liren Chen and Katia Sycara. 1998. WebMate: A personal agent for browsing and searching. In *Proceedings of the International Conference on Autonomous Agents*. 132–139.
[13] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, Vol. 60.
[14] Jan De Leeuw and Sandra Pruzansky. 1978. A new computational method to fit the weighted Euclidean distance model. *Psychometrika* 43, 4 (1978), 479–490.
[15] Lieven Decock and Igor Douven. 2011. Similarity after goodman. *Review of Philosophy and Psychology* 2, 1 (2011), 61–75.
[16] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2016. EventAction: Visual analytics for temporal event sequence recommendation. In *Proceedings of the IEEE Visual Analytics Science and Technology*. 61–70.
[17] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2017. Finding Similar People to Guide Life Choices: Challenge, Design, and Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 5498–5509.
[18] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2018. Visual interfaces for recommendation systems: Finding similar and dissimilar peers. *ACM Transactions on Intelligent Systems and Technology* PP, 99 (2018), 1–23.
[19] Fan Du, Ben Shneiderman, Catherine Plaisant, Sana Malik, and Adam Perer. 2017. Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1636–1649.
[20] Lyndsey Franklin, Catherine Plaisant, Kazi Minhazur Rahman, and Ben Shneiderman. 2016. TreatmentExplorer: An interactive decision aid for medical risk communication and treatment exploration. *Interacting with Computers* 28, 3 (2016), 238–252.
[21] Alexis Gabadinho and Gilbert Ritschard. 2016. Analyzing state sequences with probabilistic suffix trees: The PST R package. *Journal of Statistical Software* 72, 3 (2016), 1–39.
[22] Carlos A Gomez-Uribe and Neil Hunt. 2016. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems* 6, 4 (2016), 13:1–13:19.
[23] Aditya Gopalan and Shie Mannor. 2015. Thompson sampling for learning parameterized Markov decision processes. In *Conference on Learning Theory*. 861–898.
[24] David Gotz and Harry Stavropoulos. 2014. DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 1783–1792.
[25] Fabio Grandi. 2010. T-SPARQL: A TSQL2-like temporal query language for RDF. In *In International Workshop on Querying Graph Structured Data*. 21–30.
[26] Theresia Gschwandtner, Wolfgang Aigner, Katharina Kaiser, Silvia Miksch, and Andreas Seyfang. 2011. CareCruiser: Exploring and visualizing plans, events, and effects interactively. In *IEEE Pacific Visualization Symposium*. 43–50.
[27] Beverly L Harrison, Russell Owen, and Ronald M Baecker. 1994. Timelines: An interactive system for the collection and visualization of temporal data. In *Graphics Interface*. 141–141.
[28] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 230–237.
[29] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 241–250.
[30] Sylvia Ann Hewlett, Melinda Marshall, and Laura Sherbin. 2013. How diversity can drive innovation. *Harvard Business Review* 91, 12 (2013), 30–30.
[31] Petra Isenberg, Niklas Elmqvist, Jean Scholtz, Daniel Cernea, Kwan-Liu Ma, and Hans Hagen. 2011. Collaborative visualization: definition, challenges, and research agenda. *Information Visualization* 10, 4 (2011), 310–326.
[32] Gerald M Karam. 1994. Visualization using timelines. In *Proceedings of the ACM International Symposium on Software Testing and Analysis*. 125–137.
[33] Josua Krause, Adam Perer, and Harry Stavropoulos. 2016. Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 91–100.
[34] Bruce Krulwich. 1997. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine* 18, 2 (1997), 37–46.
[35] Eric S Lander, Lauren M Linton, Bruce Birren, Chad Nusbaum, Michael C Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409, 6822 (2001), 860–921.
[36] Srivatsan Laxman and P Shanti Sastry. 2006. A survey of temporal data mining. *Sadhana* 31, 2 (2006), 173–198.

[37] Thibault Ledieu, Guillaume Bouzille, Catherine Plaisant, Frantz Thiessard, Elisabeth Polard, and Marc Cuggia. 2018. Mining clinical big data for drug safety: Detecting inadequate treatment with a DNA sequence alignment algorithm. In *Proceedings of the AMIA Symposium*.
[38] Weizhong Li and Adam Godzik. 2006. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 13 (2006), 1658–1659.
[39] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
[40] Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2122–2131.
[41] Augusto Q Macedo, Leandro B Marinho, and Rodrygo LT Santos. 2015. Context-aware event recommendation in event-based social networks. In *Proceedings of the ACM Conference on Recommender Systems*. 123–130.
[42] Abdul Majid, Ling Chen, Gencai Chen, Hamid Turab Mirza, Ibrar Hussain, and John Woodward. 2013. A context-aware personalized travel recommendation system based on geotagged social media data mining. *International Journal of Geographical Information Science* 27, 4 (2013), 662–684.
[43] Sana Malik, Ben Shneiderman, Fan Du, Catherine Plaisant, and Margret Bjarnadottir. 2016. High-volume hypothesis testing: Systematic exploration of event sequence comparisons. *ACM Transactions on Interactive Intelligent Systems* 6, 1 (2016), 9:1–9:23.
[44] Heikki Mannila and Pirjo Ronkainen. 1997. Similarity of Event Sequences. *TIME* 97 (1997), 136–140.
[45] Heikki Mannila, Hannu Toivonen, and A Inkeri Verkamo. 1997. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery* 1, 3 (1997), 259–289.
[46] Richard O Mason. 1986. Four ethical issues of the information age. *MIS Quarterly* 10, 1 (1986), 5–12.
[47] Matthew Louis Mauriello, Ben Shneiderman, Fan Du, Sana Malik, and Catherine Plaisant. 2016. Simplifying overviews of temporal event sequences. In *CHI Extended Abstracts on Human Factors in Computing Systems*. 2217–2224.
[48] Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. 2002. On the recommending of citations for research papers. In *Proceedings of the ACM conference on Computer supported cooperative work*. 116–125.
[49] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. 2004. Semantically enhanced collaborative filtering on the web. In *Web Mining: From Web to Semantic Web*. 57–76.
[50] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. 2013. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2227–2236.
[51] Megan Monroe, Rongjian Lan, Juan Morales del Olmo, Ben Shneiderman, Catherine Plaisant, and Jeff Millstein. 2013. The challenges of specifying intervals and absences in temporal queries: A graphical language approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2349–2358.
[52] Gonzalo Navarro. 2001. A guided tour to approximate string matching. *Comput. Surveys* 33, 1 (2001), 31–88.
[53] Helen Nissenbaum. 1994. Computing and accountability. *Commun. ACM* 37, 1 (1994), 72–81.
[54] Cathy O'Neil. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group.
[55] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering* 16, 11 (2004), 1424–1440.
[56] Adam Perer and David Gotz. 2013. Data-driven exploration of care plans for patients. In *CHI Extended Abstracts on Human Factors in Computing Systems*. 439–444.
[57] Adam Perer and Fei Wang. 2014. Frequence: interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the International Conference on Intelligent User Interfaces*. 153–162.
[58] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: Visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 221–227.
[59] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 175–186.
[60] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
[61] Francesco Ricci, Dario Cavada, Nader Mirzadeh, and Adriano Venturini. 2006. Case-based travel recommendations. *Destination Recommendation Systems: Behavioural Foundations and Applications* (2006), 67–93.
[62] Simone Santini and Ramesh Jain. 1999. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 9 (1999), 871–883.
[63] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the International Conference on World Wide Web*. 285–295.

[64] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. 2012. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2431–2440.
[65] Guy Shani, Ronen I Brafman, and David Heckerman. 2002. An MDP-based recommender system. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 453–460.
[66] Upendra Shardanand and Pattie Maes. 1995. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 210–217.
[67] Ben Shneiderman and Catherine Plaisant. 2006. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization*. 1–7.
[68] Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI Extended Abstracts on Human Factors in Computing Systems*. 830–831.
[69] Richard Snodgrass. 1987. The temporal query language TQuel. *ACM Transactions on Database Systems* 12, 2 (1987), 247–298.
[70] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. 2005. Evaluating similarity measures: A large-scale study in the orkut social network. In *Proceedings of the ACM International Conference on Knowledge Discovery in Data Mining*. 678–684.
[71] Ashish Sureka and Pranav Prabhakar Mirajkar. 2008. An empirical study on the effect of different similarity measures on user-based collaborative filtering algorithms. In *Pacific Rim International Conference on Artificial Intelligence*. 1065–1070.
[72] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. 3104–3112.
[73] Melanie Swan. 2012. Crowdsourced health research studies: An important emerging complement to clinical trials in the public health research ecosystem. *Journal of Medical Internet Research* 14, 2 (2012), e46.
[74] Georgios Theocharous, Nikos Vlassis, and Zheng Wen. 2017. An interactive points of interest guidance system. In *Proceedings of the International Conference on Intelligent User Interfaces Companion*. 49–52.
[75] Georgios Theocharous, Zheng Wen, Yasin Abbasi, and Nikos Vlassis. 2018. Scalar Posterior Sampling with Applications. In *Advances in Neural Information Processing Systems*. 7695–7703.
[76] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.
[77] Katerina Vrotsou and Camilla Forsell. 2011. A qualitative study of similarity measures in event-based data. In *Symposium on Human Interface*. 170–179.
[78] Katerina Vrotsou, Anders Ynnerman, and Matthew Cooper. 2013. Are we what we do? Exploring group behaviour through user-defined event-sequence similarity. *Information Visualization* 13, 3 (2013), 232–247.
[79] Taowei David Wang, Catherine Plaisant, Alexander J Quinn, Roman Stanchak, Shawn Murphy, and Ben Shneiderman. 2008. Aligning temporal data by sentinel events: Discovering patterns in electronic health records. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 457–466.
[80] Douglas J White. 1985. Real applications of Markov decision processes. *Interfaces* 15, 6 (1985), 73–83.
[81] Paul Wicks, Michael Massagli, Jeana Frost, Catherine Brownstein, Sally Okun, Timothy Vaughan, Richard Bradley, and James Heywood. 2010. Sharing health data for better outcomes on PatientsLikeMe. *Journal of Medical Internet Research* 12, 2 (2010), e19.
[82] Krist Wongsuphasawat and David Gotz. 2012. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2659–2668.
[83] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. 2011. LifeFlow: Visualizing an overview of event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1747–1756.
[84] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2648–2659.
[85] Krist Wongsuphasawat and Ben Shneiderman. 2009. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *IEEE Symposium on Visual Analytics Science and Technology*. 27–34.
[86] Mohammed J Zaki. 2001. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning* 42, 1-2 (2001), 31–60.
[87] Emanuel Zgraggen, Steven M. Drucker, Danyel Fisher, and Robert DeLine. 2015. (s|qu)eries: Visual regular expressions for querying and exploring event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2683–2692.
[88] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. 2015. MatrixWave: Visual comparison of event sequence data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 259–268.