ABSTRACT

Title of dissertation: EXPLAINABLE RECOMMENDATION FOR EVENT SEQUENCES: A VISUAL ANALYTICS APPROACH

Fan Du
Doctor of Philosophy, 2018

Dissertation directed by: Professor Ben Shneiderman
Department of Computer Science

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 dissertation 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. 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 choices, which are more likely to lead to their desired outcomes.

This dissertation'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 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.

This dissertation 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. Results from empirical studies show that these prototypes can assist users in making action plans and raise users' confidence in following their plans. It presents case studies in three domains to demonstrate the effectiveness and safety 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.

This dissertation contributes an analytical workflow, an interactive system, and design guidelines identified in empirical studies and case studies, opening new avenues of research in explainable event sequence recommendations based on personal histories. It enables people to make better decisions for critical life choices with higher confidence.

EXPLAINABLE RECOMMENDATION FOR EVENT SEQUENCES: A VISUAL ANALYTICS APPROACH

by

Fan Du

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Ben Shneiderman, Chair/Advisor
Dr. Catherine Plaisant, Co-Advisor
Professor Neil Spring
Professor Niklas Elmqvist
Professor Hector Corrada Bravo

Dedication

To my wife, Tina.

Acknowledgments

This five-year doctoral study is an invaluable experience in my life. It has been a challenging, exciting, and rewarding adventure, and I was fortunate and grateful to receive support from many people. Without the help from my advisors, collaborators, friends, and family, this dissertation work would not have been successfully completed.

First and foremost, I would like to thank my advisors, Ben Shneiderman and Catherine Plaisant, who consistently gave me advice and freedom to pursue my research in event sequence analytics. Thank you Ben, for shaping my thinking and guiding me through countless challenges and obstacles. Thank you Catherine, for your insightful and practical feedback on every aspect of my projects. From you, I learned how to build a research project from scratch, I learned how to connect and collaborate with others, and most importantly, I learned how to always maintain a positive attitude at work and in life.

I would also like to thank my dissertation committees who have been supportive to my work: Neil Spring, Niklas Elmqvist, and Hector Corrada Bravo. Thank you Neil for helping me refine my research ideas and providing invaluable feedback on my system prototypes. Thank you Niklas, for reading my paper drafts at HCIL clinics. Your comments always made my paper submissions considerably stronger. Thank you Hector, for inspiring me to apply machine learning to my research problems. It was always refreshing to discuss with you.

This dissertation could not have been completed without the support from

many respected researchers and colleagues, who helped me with EventAction case studies. I would like to thank Sana Malik, Eunyee Koh, and Georgios Theocharous for evaluating EventAction in digital marketing, Seth Powsner, Jeff Belden, and Kenyon Crowley for trying out EventAction in healthcare, and Neil Spring, Evan Golub, and Jennifer Story for applying EventAction in student advising. I wish to extend thanks to Nan Cao, Jian Zhao, Yu-Ru Lin, Panpan Xu, Conglei Shi, David Gotz, Adam Perer, and Karlyn Beer, for collaborating with me on many interesting research projects since the very beginning of my doctoral study. I also am very grateful to Adobe Research for sponsoring my dissertation work for two years and offering me a wonderful internship to apply my knowledge to real applications.

The next acknowledgements will go to HCIL. I was lucky to spend five years in the lab and have received warm support from wonderful HCIL friends, colleagues, and alumni: Karthik Badam, Deok Gun Park, Megan Monroe, Christopher Imbriano, Meethu Malu, Krist Wongsuphasawat, David Wang, Jennifer Preece, Jon Froehlich, and Leah Findlater. The five years have passed so fast with fun and laughter.

Finally, I cannot thank enough my family for their consistent support, encouragement, and love throughout my life. I also wish to give special thanks to Huamin Qu, who introduced me into the world of information visualization.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

"History does not repeat itself, but it rhymes."

–Mark Twain

Recommender systems are being 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 dissertation 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. 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 choice, which are more likely to lead to their desired outcomes.

Imagine the following scenario (illustrated in Figure 1.1): I am a student at

Figure 1.1: An illustration of the student advising scenario.

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, I look at those who became professors, review the recommendations, and discuss together 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 of this dissertation 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 [13] and T-SPARQL [14]) 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 not be similar to the seed record in aspects outside the query rules. Besides, precisely formulating temporal queries remains difficult and time-consuming for many domain experts. My 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 [7] uses a network structure to aggregate similar event sequences into progression pathways and summarizes the pathways' possible outcomes. TreatmentExplorer [8] provides a novel graphic interface for presenting the outcomes, symptoms, and side effects of treatment plans. CoCo [10] 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. My 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, my approach enables users to explore multiple outcomes.

1.2 Contributions

This dissertation'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 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.

This dissertation 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. Results from empirical studies show that these prototypes can assist users in making action plans and raise users' confidence in following their plans. It presents case studies in three domains to demonstrate the effectiveness and safety 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. This dissertation opens new avenues of research in explainable event sequence recommendations based on personal histories and enables people to make better decisions for critical life choices with higher confidence.

The concrete contributions of this dissertation are:

- A systematic analytical workflow for event sequence recommendation that will be applicable in diverse applications (Figure 1.2).

- An interactive prescriptive analytics system and user interfaces to assist users in making action plans and to raise users' confidence in the action plans (Figure 1.3), and the integration of an automatic sequence recommendation algorithm to reduce users' effort in using the system.

- Empirical studies of interface components and case studies in three domains, including education, marketing, and healthcare, that provide evidence of the effectiveness of generating event sequence recommendations based on personal histories.

- Design guidelines for the construction of event sequence recommendation user interfaces and usage guidelines for mitigating the ethical issues in dealing with personal histories.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows: Chapter 2 discusses existing techniques and software tools that can contribute to generating and presenting recommendations of event sequences; Chapter 3 describes the design and evaluation of *PeerFinder* (Figure 1.4), a visual interface that enables users to find and explore records that are similar to a seed record; Chapter 4 introduces a novel hierarchical visualization (*LikeMeDonuts*, Figure 1.5) that provides an overview of a group of similar records with a flexible hierarchy of criteria values, similarity encoding, and interactive support for trimming the group; Chapter 5 introduces the workflow (Figure 1.2), user interface (*EventAction*, Figure 1.6), and automatic algorithm for event sequence recommendation; Chapter 6 reports on case studies that demonstrate the applications of my research to solve real problems in three different domains; Finally, Chapter 7 describes the design guidelines and usage guidelines produced through my studies, summarizes the contributions of this dissertation,

discusses promising future directions, and gives closing remarks.

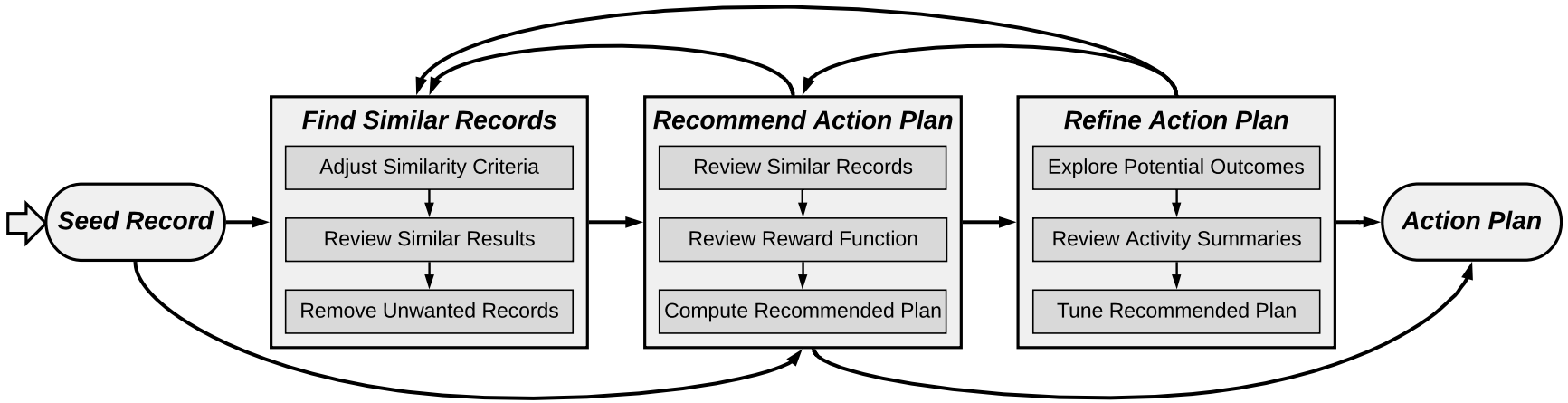

Figure 1.2: *Workflow* for event sequence recommendation: The typical workflow starts from a seed record and the first step is to find a group of similar archived records. After submitting the similar records and the reward function, a recommendation model will be computed and users can review a recommended action plan. Then, users can further refine the plan by directly editing the plan using the activities of similar records as a reference, adjusting the reward function to receive an updated recommendation, or refining the similar records.

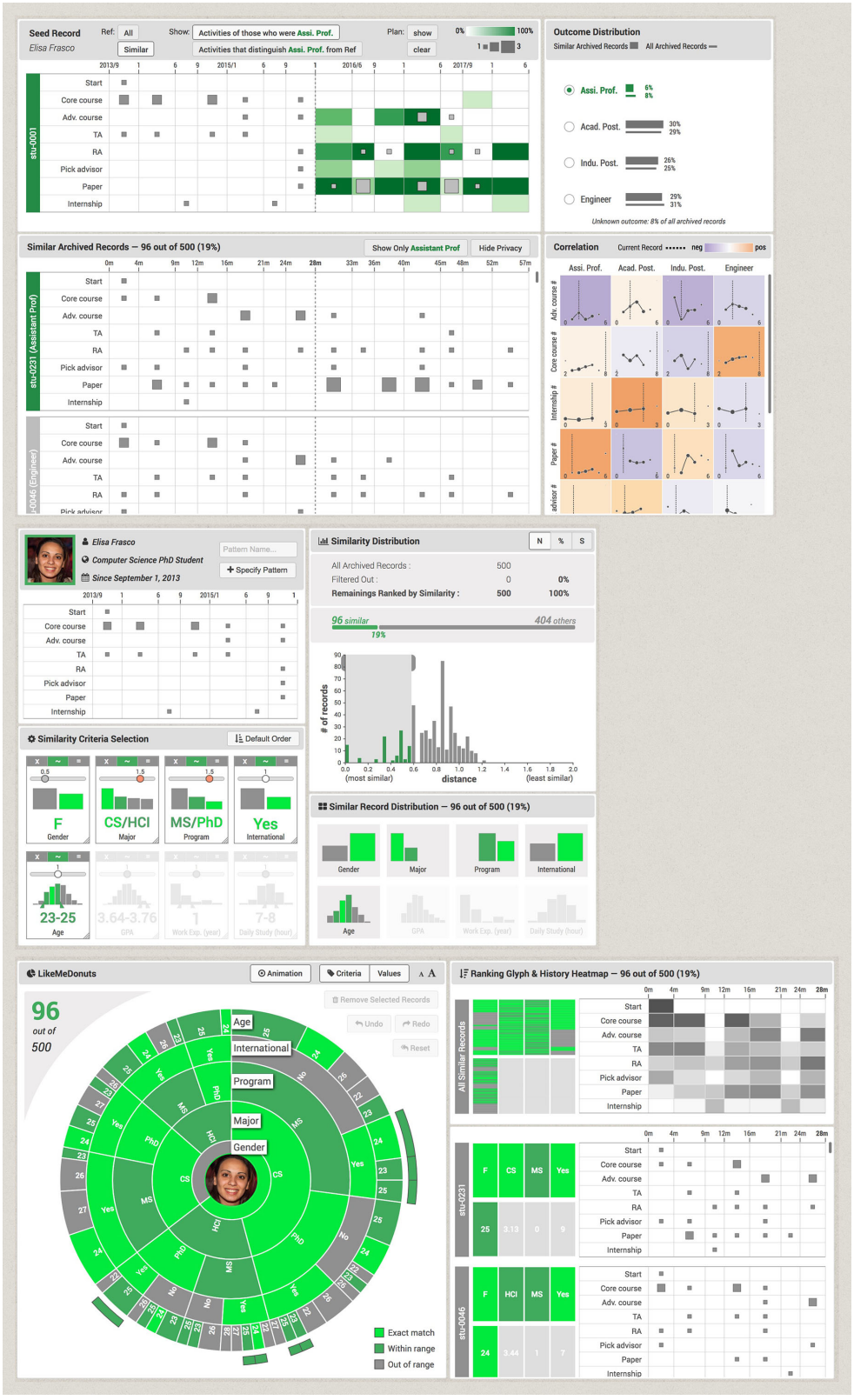

Figure 1.3: The user interface of the final design of EventAction for supporting a seamless analytical workflow for making action plans to achieve the desired outcome.

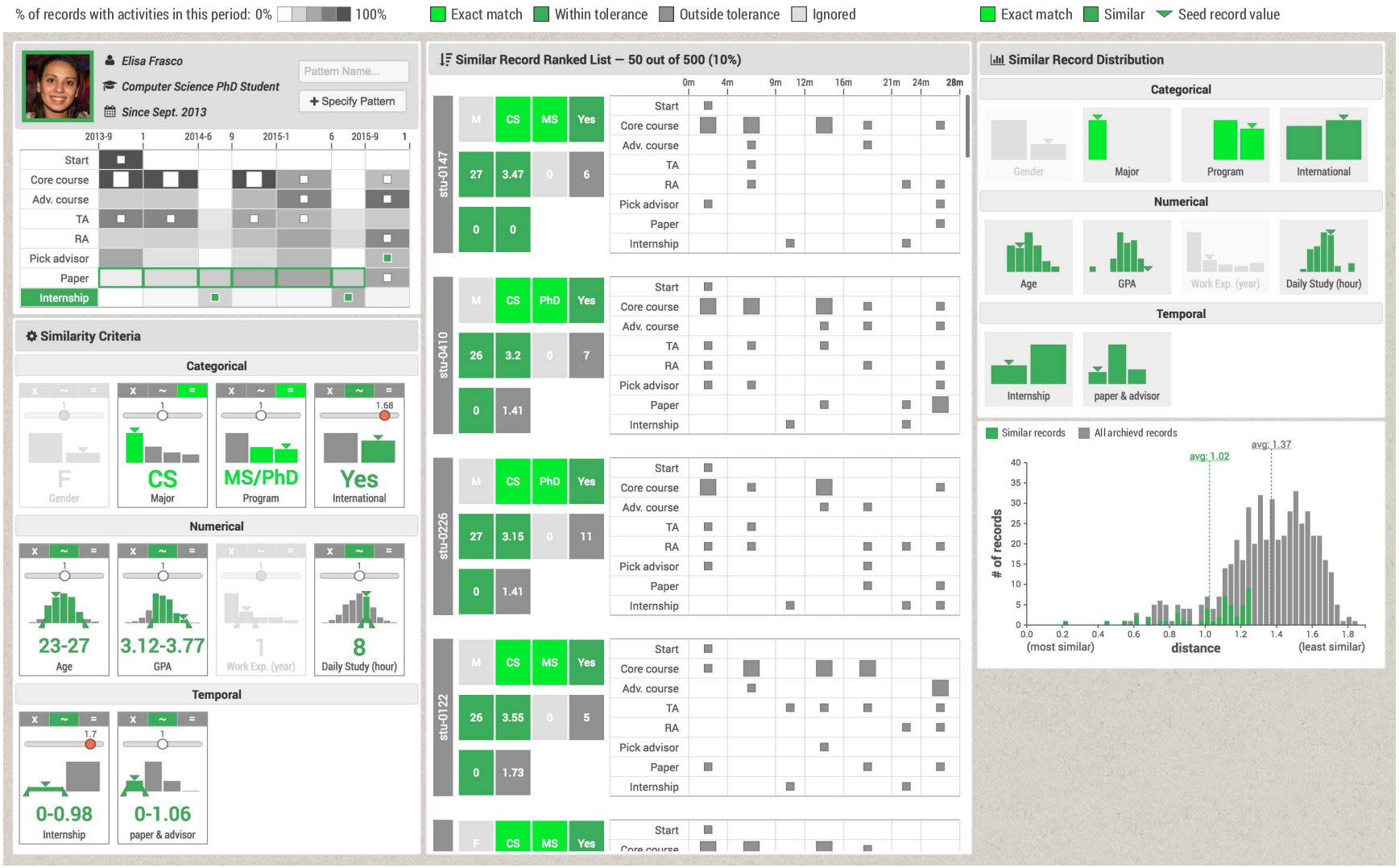

Figure 1.4: *PeerFinder*: A visual interface that enables users to find and explore records that are similar to a seed record. To encourage engagement and inspire users trust in the results, PeerFinder provides different levels of controls and context that allow users to adjust the similarity criteria. It also allows users to see how similar the results are to the seed record. Intermediate results are displayed and users can iteratively refine the search.

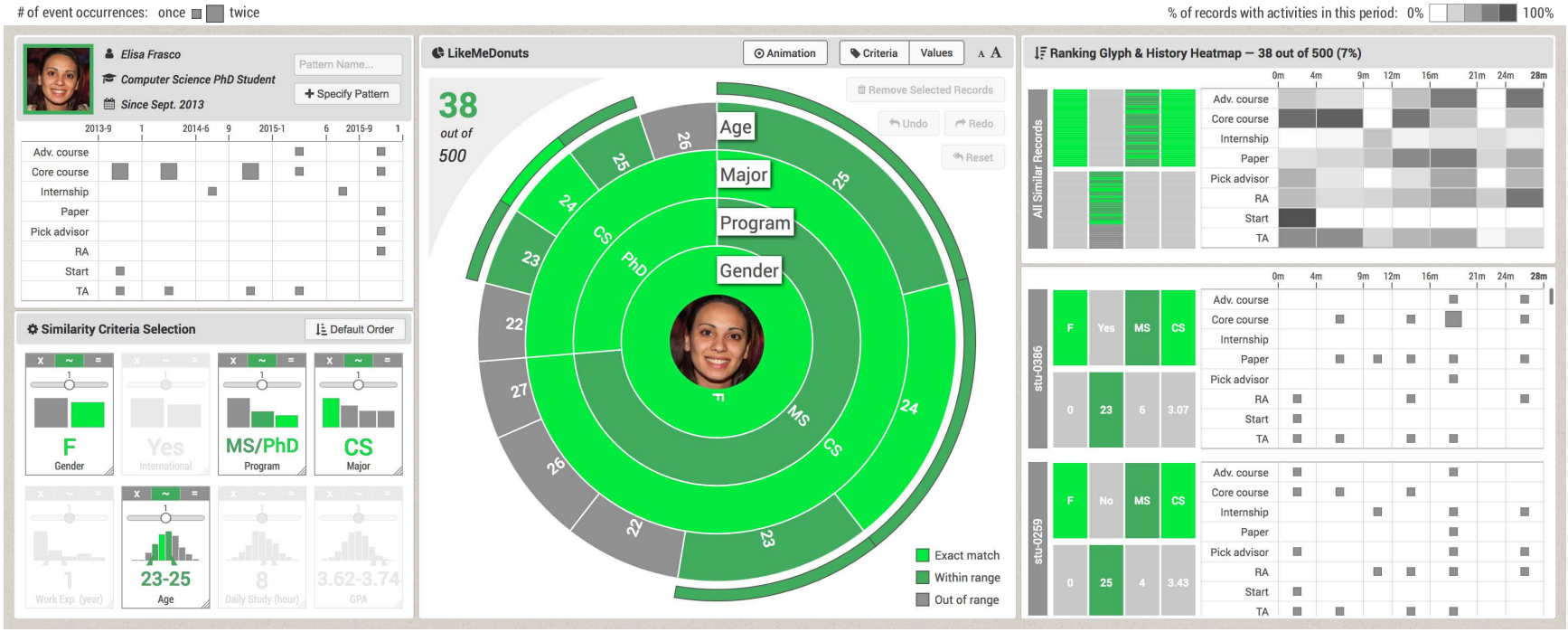

Figure 1.5: *LikeMeDonuts*: A novel hierarchical visualization that provides an overview of a group of similar records with a flexible hierarchy of criteria values, similarity encoding, and interactive support for trimming the group.

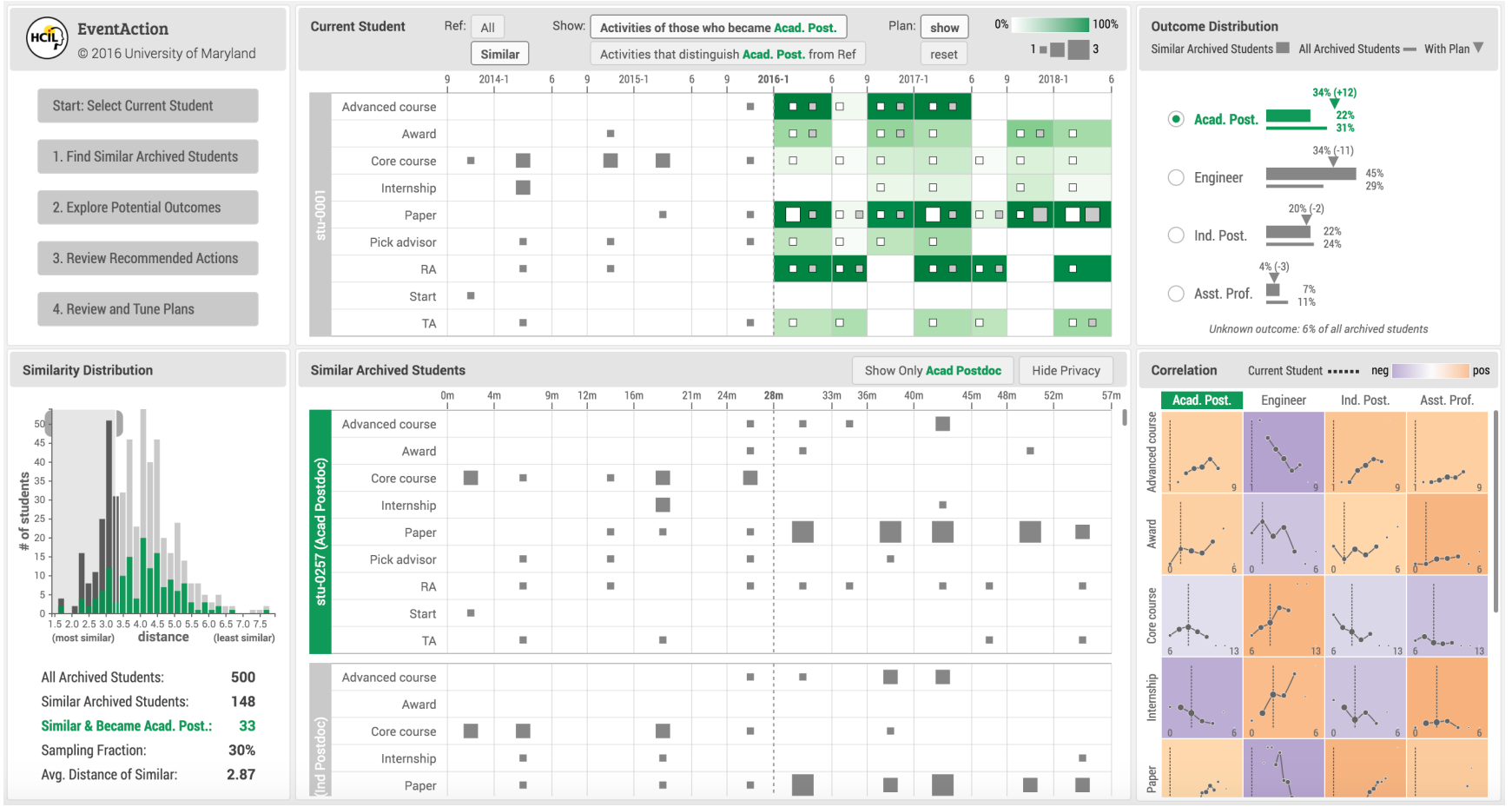

Figure 1.6: *EventAction*: A visual analytics approach to (1) identify similar records, (2) explore potential outcomes, (3) review recommended event sequences that might help achieve the users goals, and (4) interactively assist users as they define a personalized action plan associated with a probability of success.

Chapter 2: Background and Related Work

I summarize existing techniques and software tools that can contribute to my goal of enabling users to generate recommendations of event sequences that might lead to their desired outcome. My work is particularly inspired by previous research on recommender systems, similarity measures, event sequence analysis, collaborative visualization, 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 [15].

2.1.1 Recommendation Techniques

Previous work identified four major classes of recommendation techniques [16]. The two most popular ones are content-based, which recommends items similar to what the users liked in the past [17], and collaborative filtering, which finds other users with similar tastes and recommends items they liked to the current

user [18–20]. When large-scale user profiles are available, demographic techniques can be used to generating user-specific recommendations based on common patterns in the population [21]. 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 [22, 23].

In practical applications, multiple recommendation techniques are often combined to encourage the strength and diminish the weakness [24, 25]. 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 [26] and users' daily activity patterns should be considered when recommending social events [27].

2.1.2 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 [29] 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. [30] 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 [31] 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 [19, 27, 32].

2.1.3 Opportunities

My recommendation approach extends the collaborative filtering technique since I 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, my dissertation 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, I 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 [33]. For example, in bioinformatics for gene sequence alignment [34] or protein clustering [35], in linguistics for approximate string matching [36] or text categorization [37], in computer vision for face recognition [38], and in healthcare for identifying similar patients [39, 40].

2.2.1 Multidimensional Data

Data scientists investigated how to measure the similarity between two multidimensional data cubes. For example, Baikousi et al. [41] conducted user studies to explore various distance functions to identify the preferred measurement between values of a dimension and between data cubes. Spertus et al. [42] present 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 [43] extensively studied different similarity measures for online user profiles and discover that no single similarity measure could produce the best results for all users. They suggest using different similarity measure for different users.

I 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. My interviews confirmed that choices of similarity measures rely on users' preferences and analysis goals, and my user studies revealed that providing

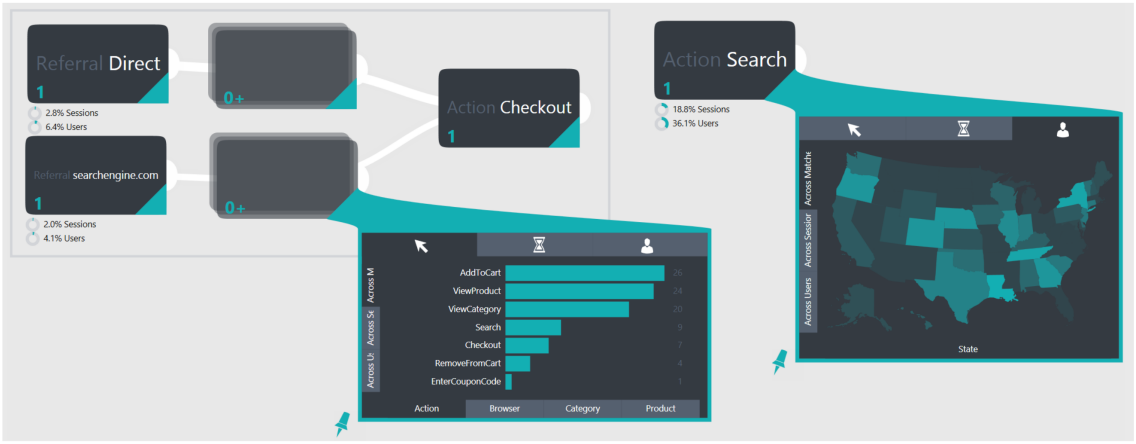

Figure 2.1: (s|qu)eries [1] provides a visual interface for specifying queries on event sequence data based on regular expressions.

controls and context will increase users' engagement and trust in similarity search results.

2.2.2 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 [13] and T-SPARQL [14]) 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 [1] (Figure 2.1), COQUITO [2] (Figure 2.2), and EventFlow [3]) (Figure 2.3).

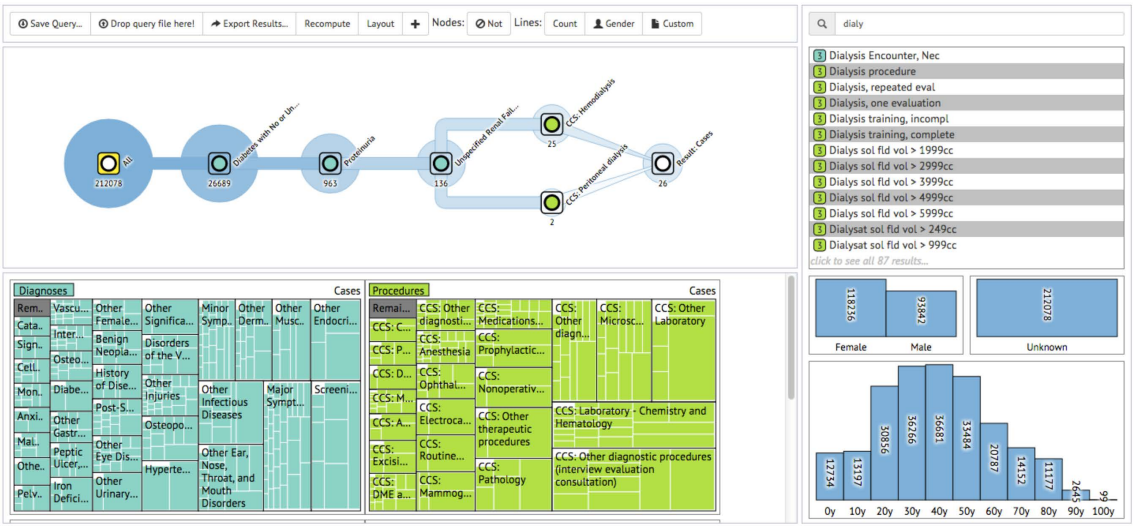

Figure 2.2: COQUITO [2] provides a visual interface that assists cohort construction with temporal constraints. Intermediate results are displayed so users can iteratively refine a query.

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 [45] introduces a similarity

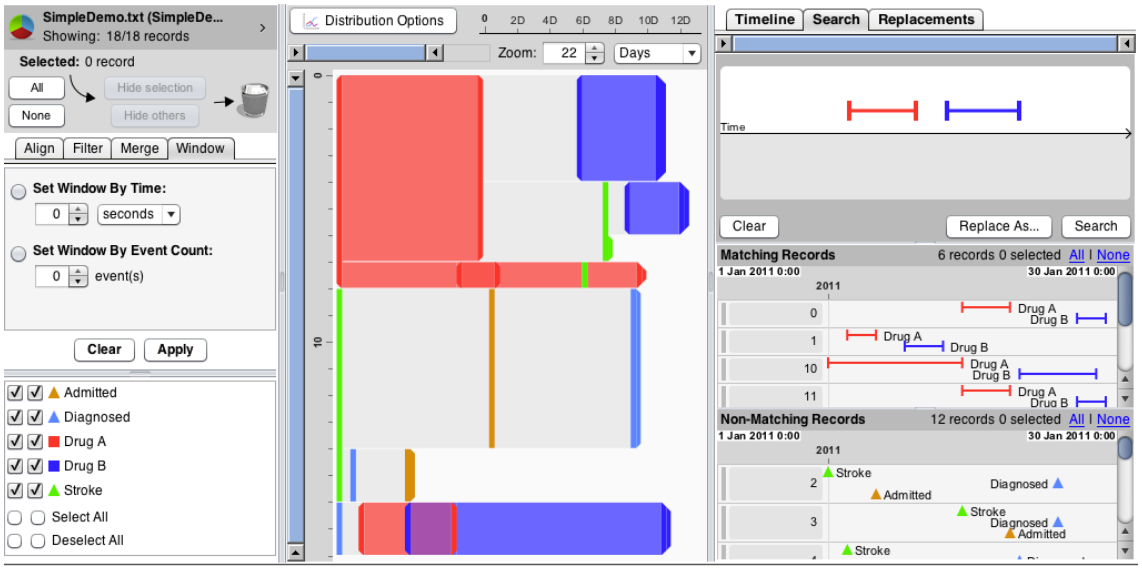

Figure 2.3: EventFlow [3] presents a visual query language that allows users to draw the desired sequence of event relationships. Results are displayed as detailed timelines and summarized in an aggregated overview.

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 [46, 47] 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 [48].

My 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.

2.3.1 Visual Representations

Starting with LifeLines [4, 5] (Figure 2.4), early research on event sequence visualization focuses on depicting the medical history of a single patient (e.g., Bade et al. [49], Harrison et al. [50], and Karam [51]). These tools allow users to visually inspect trends and patterns in a record by showing detailed events. LifeLines2 [6] (Figure 2.5) extends this approach to multiple records but do 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 [52] introduces a method to aggregate multiple event sequences by combining them into a tree structure on an alignment point. Likewise, OutFlow [7] (Figure 2.6) combines multiple event sequences based on a network of states. EventFlow [53] extends LifeFlow's concept to interval events and introduces simplification strategies to deal with large data volumes and pattern variety [54]. DecisionFlow [55] provides supports for analyzing event sequences with larger numbers of categories.

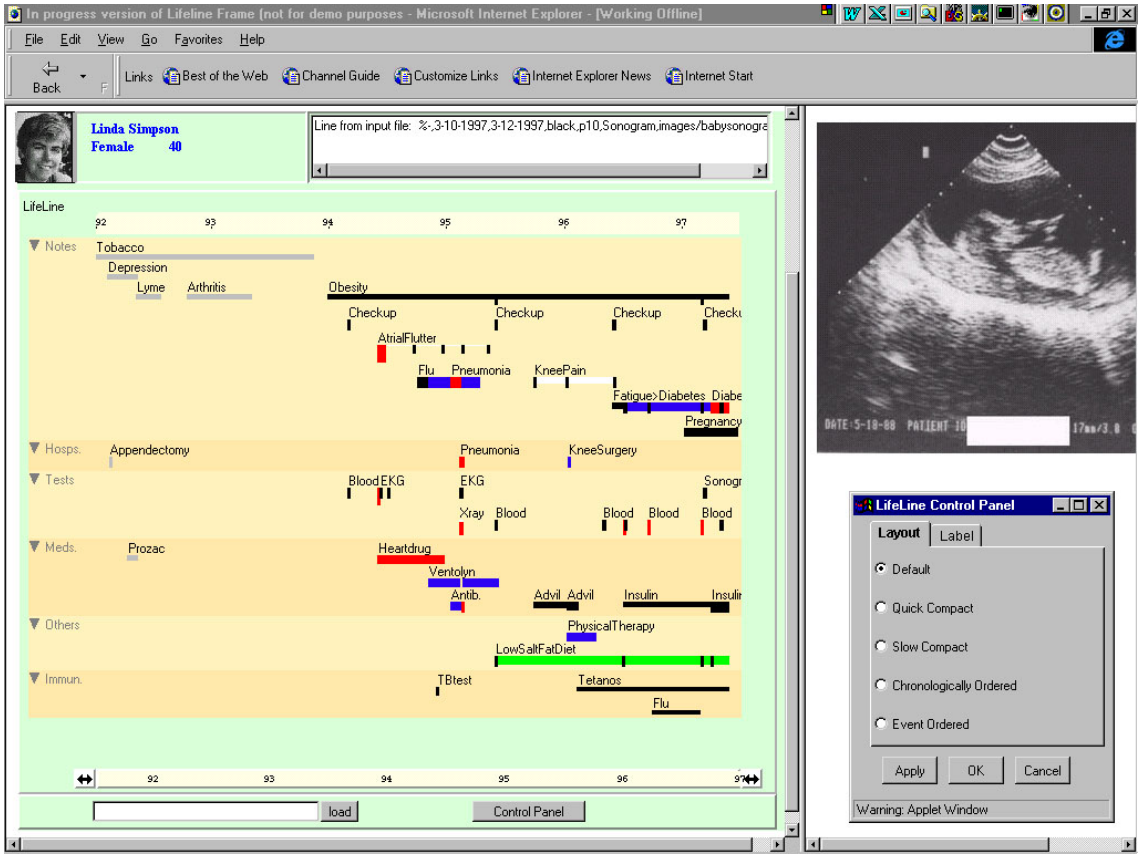

Figure 2.4: LifeLines [4,5] provides a visual interface for showing the medical history of a single patient.

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

2.3.2 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 [56]. Many techniques have been developed to support this task and the main challenge is that a combinatorially explosive number of intermediate sub-

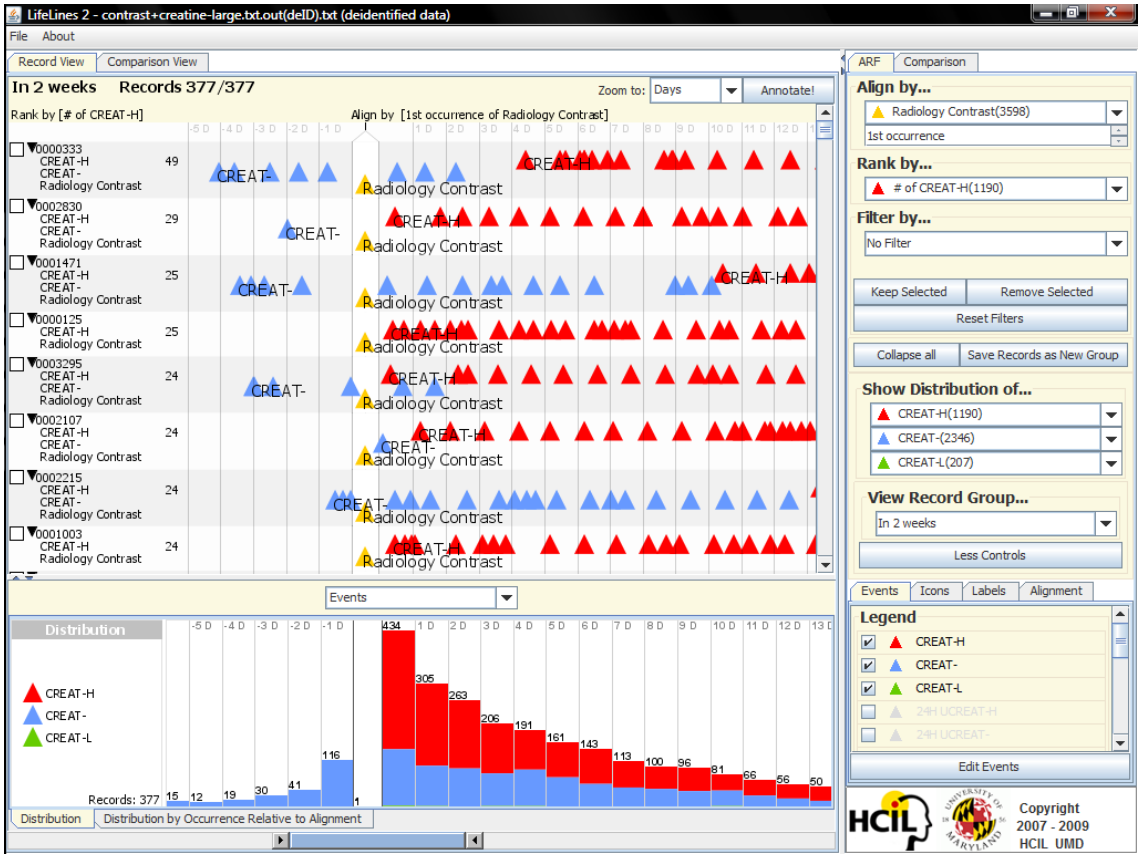

Figure 2.5: LifeLines2 [6] supports showing multiple records on the same display in a stacked manner.

sequences need to be examined. Early work mainly focused on developing efficient and automatic algorithms. Apriori-like [57,58] 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 [59] and SPADE [60] reduce the number of data scans, and SPAM [61]

Figure 2.6: OutFlow [7] uses a network structure to aggregate similar event sequences into progression pathways and summarizes the pathways' possible outcomes.

uses a bitmap representation to encode the event sequences and accelerates the mining computations with bitwise operations. Recently, Perer and Wang [62] 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 my dissertation, I will explore frequent sequence mining techniques and apply them in the system.

2.3.3 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 [7] (Figure 2.6) 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 [63], allows doctors to analyze treatment plans and their outcomes for patients with certain clinical conditions. TreatmentExplorer [8] (Figure 2.7) provides a novel graphic interface for presenting the outcomes, symptoms, and side effects of treatment plans. CareCruiser [9] (Figure 2.8) enables doctors to retrospectively explore the effects of previously applied clinical actions on a patient's condition. CoCo [10] (Figure 2.9) 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 [11] (Figure 2.10) 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. My 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

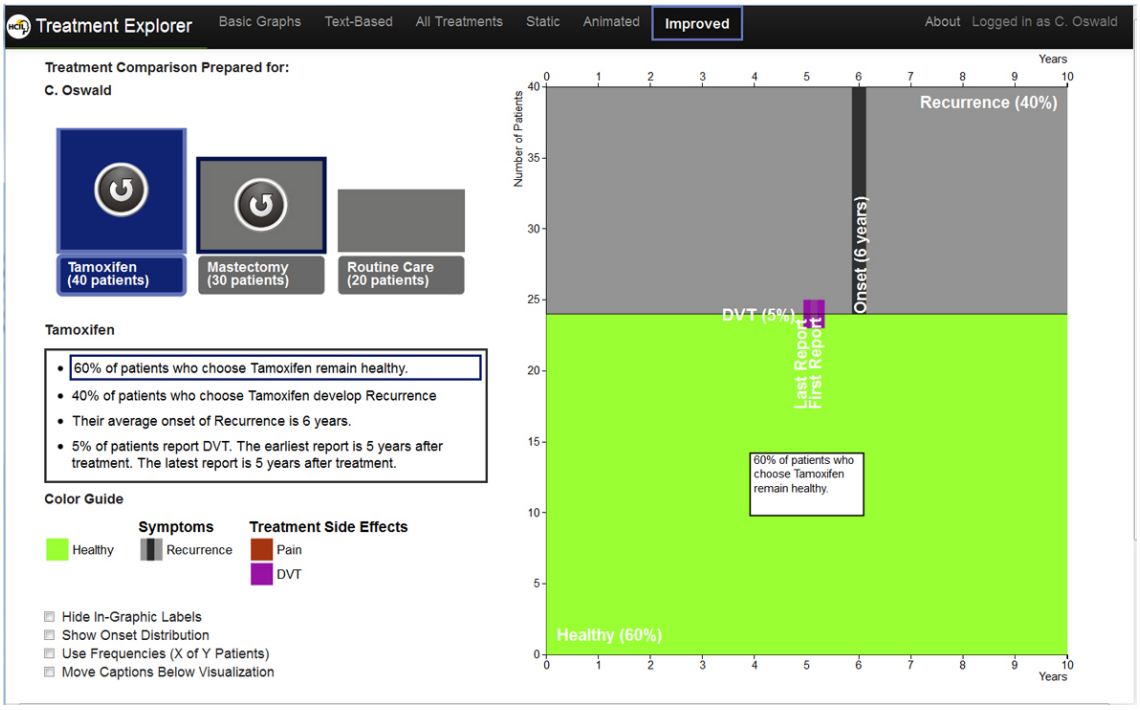

Figure 2.7: TreatmentExplorer [8] visually presents the outcomes, symptoms, and side effects of treatment plans.

addition, while most existing tools assume a binary outcome, my 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 life, the same technology may also raise ethical issues such as threatening our privacy or providing inaccurate information that mislead our decisions. Mason [64] 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 [65] introduces the

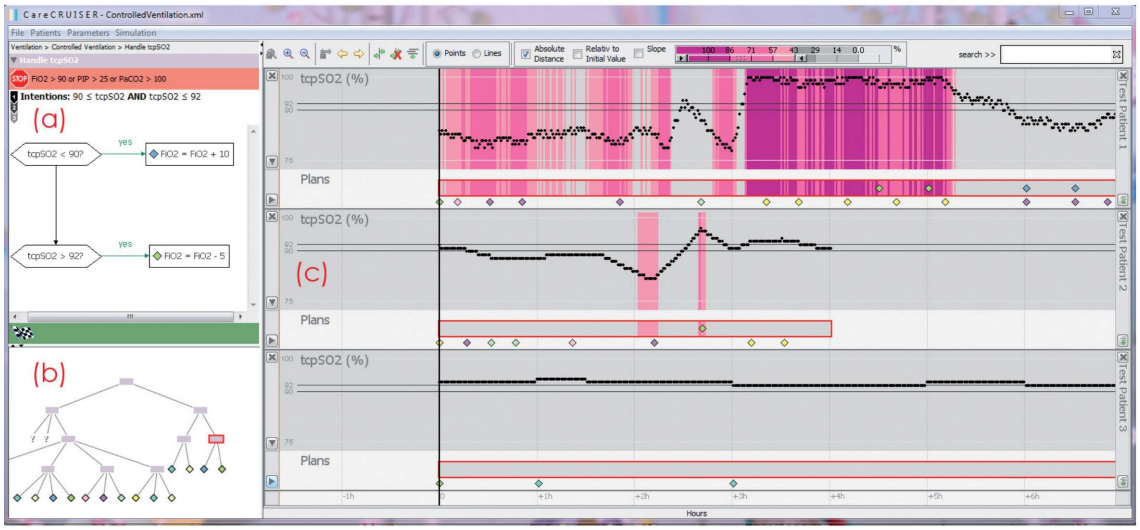

Figure 2.8: CareCruiser [9] supports exploring the effects of previously applied clinical actions on a patient's condition.

concept of accountability in computing to ensure that harms and risks caused by technology can be answered and handled.

One main source of ethical issues is the bias in computer systems, which can be further categorized into three groups: preexisting, technical, and emergent [66]. Specifically, preexisting biases originate in social institutions or personal biases of individuals who design the system, and in contrast, technical biases typically relate to limitations of computer hardware and software. After the system has been built, emergent biases may occur as it encounters situations that have not been considered in the design, most often when the usage of the system extends or the context of use shifts.

In my dissertation, by working with real users and domain professionals, I will study the ethical issues in dealing with personal histories. Specifically, I will investigate (1) what the potential biases are in using histories of similar others to provide recommendations, (2) what the potential dangers are in allowing advisees

Figure 2.9: CoCo [10] enables systematic exploration of event sequence comparisons. Given two groups of records, it detects their differences in the composition of the event sequences.

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. I will discuss these ethical issues and propose possible solutions.

2.5 Summary

This chapter discussed previous work in various related research topics, including recommender systems, similarity measures, event sequence analysis, and ethical issues in information systems. These techniques, software tools, and theories can contribute to my goal of enabling users to generate recommendations of event sequences that might lead to their desired outcome. My dissertation will contribute a systematic analytical workflow and an interactive prescriptive analytics system for event sequence recommendation. My empirical studies and case studies will produce

Figure 2.10: MatrixWave [11] allows visually compare two set of events sequences by creating a matrix visualization that shows the differences at each step.

design guidelines for the construction of event sequence recommendation user interfaces and usage guidelines for mitigating the ethical issues in dealing with personal histories.

Chapter 3: Finding Similar People to Guide Life Choices: Challenge, Design, and Evaluation

People often seek to use examples of similar individuals to guide their own life choices. For instance, patients may want to receive the treatments that work for others with similar physical conditions and disease symptoms, or new students may wish to follow the trajectory of former graduates who had similar backgrounds and academic performances and ended up with a successful career. In the era of big data, where electronic health records and electronic student records are commonplace, exploring the data of similar individuals to receive advice on life choices and foresee potential outcomes is becoming possible. However, finding the records of similar individuals from databases is an important yet difficult step, often overlooked or existing in some analytical applications only as a black-box process [7, 8, 67].

Imagine a patient suffering from a knee injury who wants to understand if people like her chose surgery first then physical therapy or a more conservative treatment, and wants to know how long before they return to normal use of their knees. But what data should be used as evidence for people like her? As a light-weight woman in her thirties, will she trust results based on data from older women? From strong athletes? From those with prior knee injuries? Or with several unre-

lated medical conditions? To narrow the results, more information from the medical record could be used to tailor the set of similar patients, such as the degree of everyday physical activities and previous knee conditions. Specifying such a query using standard tools is incredibly complex as a large number of rules need to be specified, and since every person is unique, the result set of specific and complex queries is likely to be empty.

To understand users' needs, I reflected and built on experience accumulated from working with case study partners (medical researchers, doctors, marketing and transportation analysts, etc.) for more than a decade while developing tools and interfaces for the exploration of personal records. Searching for similar records was requested by many users. My long-term goal is to support prescriptive analytics interfaces that guide users as they make plans informed by the history of similar people [7, 8, 67–69]. Searching for similar records is the focus of this dissertation.

After summarizing the challenges in finding similar people, I report on the results of 13 interviews that informed my design effort. I implemented PeerFinder[1], a visual interface that enables users to find and explore records that are similar to a seed record (either their own record or the record of a person they intend to counsel). PeerFinder uses both record attributes and temporal event information. To encourage engagement and inspire users' trust in the results, PeerFinder provides different levels of controls and context that allow users to adjust the similarity criteria. It also allows users to see how similar the results are to the seed record. Intermediate results are displayed and users can iteratively refine the search.

[1]This work was published at ACM CHI 2017 [70].

My contributions include:

- A clarification of the challenges in finding similar people to guide life choices and a need analysis with 13 interviews.
- A flexible prototype, PeerFinder, which allowed us to explore different levels of controls and context, and interface styles to refine the results.
- The results of a user study with 18 participants and 4 expert reviewers comparing three interface configurations.

3.1 Challenges

Every person is unique, and finding similarities between individuals is a multifaceted and subjective process. This dissertation focuses on similarity in the context of making critical life choices (and not other uses such as eliminating duplications, searching for criminal activity, or finding job applicants).

3.1.1 Trust in the Evidence Contained in the Results

Making life choices based on data found in similar records takes a leap of faith. It implies that users are confident that the found records are similar enough to them to provide personalized evidence to guide their choices, and that decisions that were optimal for similar records will also be optimal for them. This confidence may be based on (1) trust in the source of the data and algorithm (e.g., results coming from one's doctor or NIH may be trusted more than those coming from an unknown source), (2) previous experience (e.g., once results have been found useful,

the next result may be more likely to be trusted), and (3) understanding how the results were obtained (e.g., looking under the hood and being able to adjust the search parameters) [31, 71, 72]. Increased knowledge may also (appropriately) lead to lower trust when users realize that the results are not really very similar to the seed record [73].

3.1.2 No Natural Computable Distance Measure

Electronic records of personal histories (e.g., patients, students, historical figures, criminals, customers, etc.) consist of multivariate data (e.g., demographic information) and temporal data (time-stamped events such as first diagnosis, hospital stays, interventions) where each event belongs to a category. Intuitively, we can consider a record that is identical to the seed record to be the most similar while a record with all opposite attribute values and no common events can be seen as the most dissimilar, but defining a nuanced similarity measure to rank records by similarity is challenging.

The similarity between numerical values (e.g., age or weight) can be easily assessed by standard distance functions and normalized. Ordinal values also lend themselves to such distance (e.g., student letter grades), but categorical values pose problems. Sometimes the distance between values can be estimated using a standard hierarchical structure, e.g., the ICD-10 codes [74] allows a distance measure between diseases to be computed. However, there are no natural distance measures for categorical attributes in general, such as between races or academic disciplines.

Moreover, temporal events add enormous complexity to the similarity measurement: not only are there no natural distance metrics between event categories but there is no generally accepted method to rank differences in sequence patterns. Specifically, what should be the "distance" created by a missing event or a reordering of events?

Nevertheless, it is possible to define an initial similarity for each pair of records as a weighted composite of scores arbitrarily set for all individual measures and possible differences. Hundreds of arbitrary decisions have to be made, but users may be able to adjust those parameters for specific applications.

3.1.3 The Subjective Nature of Similarity

While there is no natural numerical distance between people, patients and students express very strong opinions about records being similar or dissimilar to them based on how they identify or not with the other person, making the notion of similarity very subjective. How people perceive similarity depends on their preferences, experiences, and beliefs, and has been dismissed by some as a slippery notion [75]. Educators may see students of different majors as absolutely dissimilar. Doctors may see as the most similar the patients that are taking exactly the same combination of drugs.

3.1.4 Similar for Which Purpose?

How people evaluate similarity is affected by their goals. Someone looking for medical guidance will most likely ignore the similarity of education or place of

residence. I identified the following possible use of a similarity search:

- Compute outcome measures, e.g., to estimate the chance of developing a disease or achieving a desired goal. Here a large number of similar records are needed, and knowledge of which criteria influence the outcome will guide the similarity judgment. Physicians may know that having had children affects certain types of cancer but patients may not. Students may only consider publication activities to estimate the likelihood of getting a postdoc position.
- Identify stories to motivate. A physician may be trying to remember the case of a similar patient who had a good outcome to encourage a patient to follow a specific treatment. Here, gender and age may contribute little to the similarity of the clinical cases, but be required to motivate the patient.
- Make plans for future actions, e.g., to define long-term treatment plans based on the outcomes of similar patients or recommend interventions to retain a customer based on the histories of similar customers. Here the records' temporal information may become more important. For example, a student seeking course planning advice will put more weight on the similarity of the sequence of classes and grades.

3.1.5 Lack of Ground Truth Benchmark Data

Well-developed research topics such as face or image recognition, document search or topic classification have a long history and ground truth datasets have been developed to evaluate results of various algorithms and a much lesser extent of

user interfaces. Even subjective judgments have been collected and aggregated. In contrast, searching for similar people to guide life choices is a new topic of research and there exists no benchmark dataset to train machine learning models or evaluate prototypes. Besides, since the data structure and perception of similarity vary among domains, it will be difficult to generalize the evaluation results gathered from one domain to others, so various benchmark datasets will be needed.

In summary, searching for similar records is technically easy using arbitrary distance measures, but similarity judgments are subjective and there is no validated measure or established ways to measure the quality of the result set before generating personalized evidence-based recommendations for life choices. Therefore, I believe that providing users with some control over the search and context information about the results is critical to building trust in the recommendations. This dissertation is a first investigation into the design space of a new research area: personalized search for similar personal records.

3.2 Informing the Design

The challenges described above highlight the need to provide users with some level of control over the selection of the criteria to be used in the search. To further understand how users would want to specify which criteria to use and how to present results and context, I conducted a series of interviews.

3.2.1 Interviews

Thirteen potential users were interviewed (4 graduate students, 2 graduate advisors, 2 physicians, a start-up CEO, and 4 researchers working in healthcare or marketing). Each interview lasted approximately one hour, including a semi-structured interview and a ranking task to provoke further discussions. I asked participants about what information they might want to gather from similar records, what criteria they would want to use when searching for similar records, and what information would increase their confidence in the value of the results.

Three separate scenarios were used. A student advising scenario asked participants to imagine a setting where an advisor is meeting with a current student to make plans for the year. For the healthcare scenario, I asked participants to think of a doctor working with a patient to make a treatment plan. For marketing, I asked the participants to imagine that they were designing a series of interventions (e.g., calls, ads, or coupons) to retain an important customer, and could look for similar customers to inform their intervention design. Each participant chose one or two scenarios according to their backgrounds. While most participants could easily identify with the student and healthcare situations, the marketing scenario was used only by three participants. They could assume both user roles: the person expecting to receive guidance or the person hoping to provide guidance to others.

I asked the participants to discuss (1) what they would hope to learn from the data of similar records, (2) what criteria they wanted the tool to consider in the similarity search, and (3) what information they would need to determine if the

results were similar enough to provide personalized evidence. I told participants to assume that data privacy concerns had been resolved (e.g., only aggregate data would be available if access to details had not been granted).

After a period of open-ended discussion, participants were provided with six printed records, among which one was assigned as the seed and the other five were archived records being searched. Participants were encouraged to think aloud as they tried to rank the archived records by similarity to the seed record, and to describe the criteria they considered in the comparison, the difficulties they faced, and any supports they wanted from a visual interface to complete such task.

3.2.2 Results

I summarize the results and present my findings.

3.2.2.1 What to Learn from Similar Records

In all three scenarios, participants confirmed the expected uses, in particular, the prediction of outcomes. For example, students wanted to know what jobs similar students got after graduation and their salaries; marketing researchers wanted to know the likelihood of a promotion link being clicked. In addition, participants also asked for estimating the effect of an action on the future of the seed record (i.e., "what if" analysis, a simplified action plan recommendation). For example, a student wanted to test if taking an internship in the last year would increase her likelihood of getting a job at Google, and an advisor wanted to answer students

asking if taking an extra class in the next semester might drop the GPA, or if giving up a difficult class would delay graduation. A student stated that *"the information I know about my peers would definitely help me make better decisions."* Both advisers and physicians commented that they often used examples from similar records to tell motivational stories to their advisees or patients, but that it is difficult to remember those similar cases.

3.2.2.2 Similarity Criteria

Participants responded on average with 11 criteria ($SD = 3.92$), using both record attribute criteria and temporal criteria. Record attribute criteria included categorical values (e.g., gender, nationality, major, research topic, diagnosed disease, or membership tier), and numeric values (e.g., age, weight, height, family income, number of chronic problems, or company size). Temporal criteria included the time between events (e.g., between pick advisor and publication, between two painful episodes, or between sending advertisements and clicking on the promotion link), and the pattern of event occurrences (e.g., a change in the number of publications over time, lose weight and then get sick, or search for a product online and then purchase in the store). Most temporal criteria were stated in general terms (e.g., recently, in the past) with some exceptions in the medical domain, where well-defined, specific temporal patterns were mentioned.

Participants did give examples of criteria which should be ignored (e.g., women are rare in Computer Science so a female participant wanted that criterion to be

ignored). Users may also want particular time periods to be ignored as well (e.g., a school semester when the student was ill).

Some criteria were cited as being more important than others, but in many cases, participants were uncertain about how distinguishable a criterion was for the population or how relevant a criterion was for the knowledge they wanted to gain from the similar records. For example, a student advisor said: *"I am sure about certain criteria but not confident about many others. I want to use the tool to decide if a factor is important in the context of my analysis goal."* All participants mentioned their criteria depend on intended use of the similar records. A physician stated *"gender is important for finding similar patients with breast cancer but does not matter for hypertension or diabetes,"* another said *"they are similar for a purpose."*

A common method used to select criteria was to identify unique characteristics of the seed record. For example, a student may have changed advisors three times in a year, or a patient may be uninsured and cannot afford expensive treatment plans. Participants wanted the system to highlight those unique characteristics.

3.2.2.3 How to Evaluate the Similar Records

The participants proposed five possible strategies for reviewing the results and determining if they are actually similar enough to the seed record: *Sample inspection*, inspecting individual records, especially the most and least similar ones. *Difference between records*, reviewing differences between the seed record and individual similar

records. *Distributions*, reviewing histograms of the values of each criterion among similar records. *Statistical information*, reviewing the number of records in the result, the weight of each criterion, and the statistics for each criterion (e.g., min, max, mean, variance). *Context*, comparing and contrasting the set of similar records to the entire population. A student described his reason for choosing such reviewing strategies: *"I picked the criteria, so I just need to confirm if the results reflect my choices."*

3.2.2.4 System Design Needs

Based on my initial analysis and participants' suggestions, I propose a list of five design needs.

N1. *Dynamic criteria specification:* To see and adjust which criteria are used—or not, and limit acceptable tolerance.

N2. *Criteria prioritization:* To assign weights to different criteria and highlight criteria with higher importance.

N3. *Uniqueness identification:* To receive assistance in identifying unique characteristics of the seed record compared to all archived records.

N4. *Result review:* To review statistics and distributions of the similar records and detailed information of each individual in the results (if access is granted).

N5. *Goal-driven exploration:* To explore how relevant each criterion is to their analysis goal and identify important criteria depending on that goal.

I hope that providing controls over the search process (**N1-2**) and context for the results (**N3-4**) will reduce the challenges of trust and subjectivity in finding similar records. In an attempt to bound the scope of this dissertation to a similarity search interface, the last need is not addressed because it depends entirely on the end goal of the overall application. For example, if the goal is to estimate what job is most likely to be attained by a student, the application will need to identify which criteria are correlated to the student job placements. Outcome analysis tools such as DecisionFlow [55] and CoCo [10] could be used.

3.3 Description of PeerFinder

This section describes the user interface and search algorithm of PeerFinder, a visual interface that enables users to find and explore records that are similar to a seed record.

3.3.1 Interface

PeerFinder has four coordinated views[2] (Figure 3.1): on the left is the seed record with a timeline (a) and attributes (b), which are also used for criteria control. In the center is the ranked list of similar records (c), and on the right is the overview of the similar records (d). The interface can be configured by advanced users using a control panel that adjusts the visibility of all interface components. Here I describe the *Complex* version of PeerFinder configured to provide maximum control and

[2] A demo video is available at `http://hcil.umd.edu/peerfinder`.

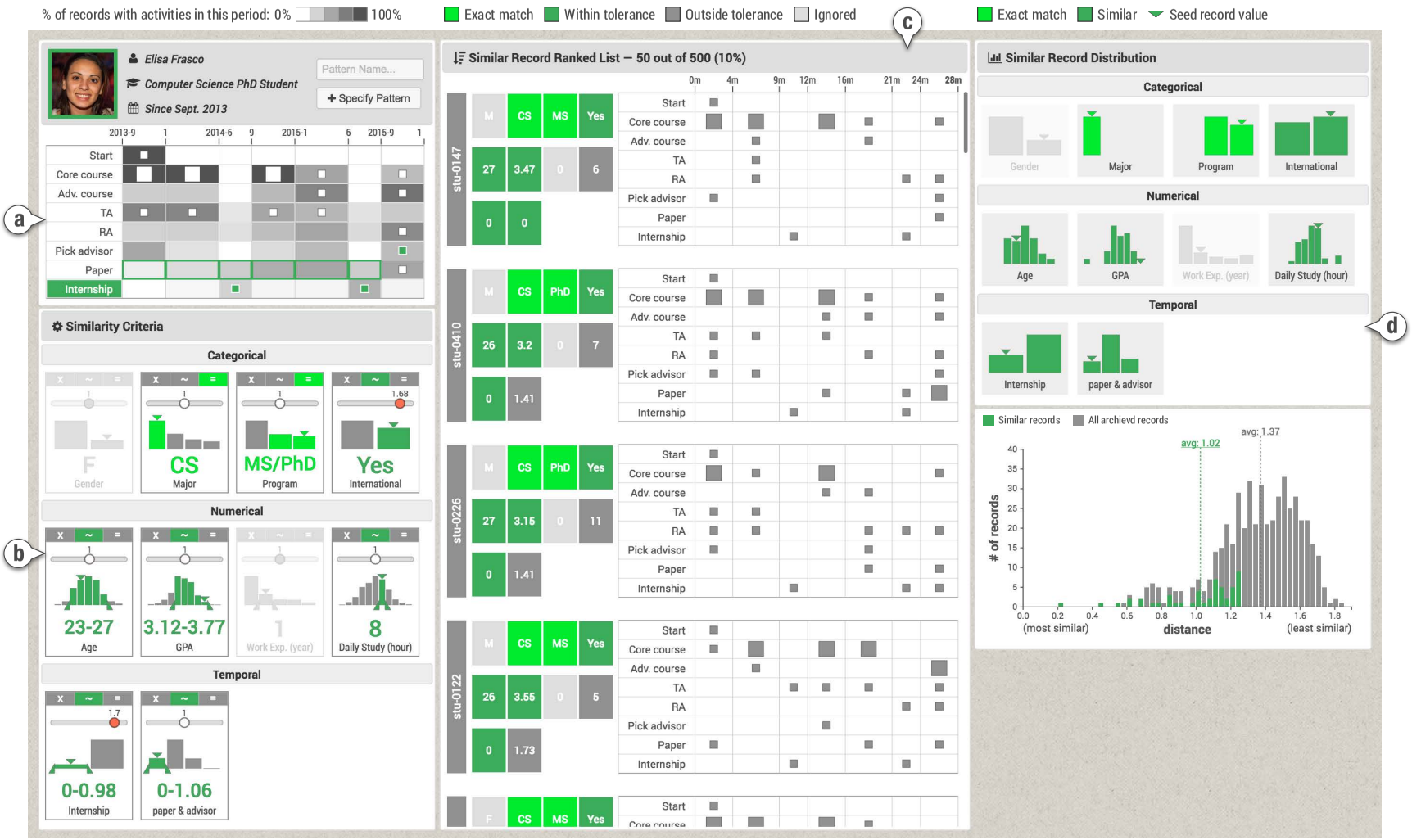

Figure 3.1: The *Complex* version of PeerFinder, showing all the criteria controls and detailed context. On the left is the seed record attributes and similarity criteria control panel (b). In the center is the ranked list of the similar records with all details (c). On the right is a summary of the results (d). The seed record is a female Ph.D. student in Computer Science. The user chooses to only keep Computer Science student in either M.S. or Ph.D. program. Tolerance ranges are specified for age and Grade Point Average (GPA). More weight is given to international students. In the timeline (a) two temporal patterns were specified and added to the criteria control panel.

context. Two simpler versions are described later.

3.3.1.1 Seed Record Timeline

A simplified timeline of the seed record is shown in a table (Figure 3.1a), where rows represent event categories and columns represent time periods. Events of the same category that occur in the same period are aggregated and shown as a square, with the size of the square encoding the number of occurrences. For students' records, time periods can be school semesters (e.g., Spring, Summer, and Fall). Advanced users can specify other time period rules based on specific data and applications. User interviews suggested that temporal criteria use only rough time periods so I chose this table-based design which simplifies the timeline while allowing users to explore how the numbers of event occurrences evolve over time.

Users can select or deselect event categories as criteria or specify temporal patterns by selecting cells in the timeline table. To provide a population overview and help users identify unique temporal patterns of the seed record, the data from all archived records are shown as a heatmap in the table background. In each table cell, the darkness of the background color encodes the percentage of records that had at least one event in this category and this period. Hovering on a cell shows the details.

3.3.1.2 Similarity Criteria Controls

Similarity criteria are displayed in three groups (Figure 3.1b): categorical (e.g., gender or major), numerical (e.g., age or GPA), and temporal. Categorical and numerical criteria are automatically defined based on the available record attributes. Temporal criteria are added when a pattern has been specified on the timeline (e.g., having an internship every summer). Each criterion is represented by a rectangular glyph showing its name and context information (i.e., the value of the seed record attribute and distribution of all archived records), along with controls for tolerance range, matching rule, and weight:

Tolerance range: Users can define a tolerance range to treat multiple categorical values or a range of numerical values equally to the value of the seed record, which will increase the similarity of records with those values. For example, users may decide to treat M.S. and Ph.D. students equally, and set a value range between 3.1 and 3.7 for GPA.

Matching rule: For each criterion, users can define its matching rule by selecting among "Ignore" ($\times$), "Close Match" ($\sim$), or "Exact Match" ($=$). The default rule for all criteria is "Close Match" where records with smaller differences from the seed record will be considered as more similar and ranked higher. The results could have diverse criteria values since the ranking considers the overall difference between records. To narrow results and explicitly include or exclude certain criteria values, users can switch to the "Exact Match" rule and use the tolerance range selector to specify the criteria values that all records in the results must match (e.g., only keep-

ing Computer Science students who have more than one year of work experience). Users can also set the rule to "Ignore" if they do not want to use that criterion.

Weight: Users can give more importance to certain criteria by adjusting their weights using a slider. Increasing the weight magnifies the differences between each archived record and the seed record while small differences in that criterion become smaller. By default, all criteria have a weight of 1, which can be adjusted to any value between 0 (ignored) and 2 (doubled). The color of the round handle becomes red when the weight is high to help users locate the criteria with higher weight.

3.3.1.3 Similar Record Ranked List

Each time users add or adjust a similar criterion, PeerFinder automatically re-runs the search and shows the refined list of the top similar records (10% by default) in a ranked list (Figure 3.1c). Each row in the list represents a similar record, consisting of a record ID, values of specified similarity criteria, and a timeline of temporal events. Specifically, the criteria values are displayed in a table with the same layout as the similarity criteria control panel. Values in a green background are within the specified criteria tolerance range while those with a gray background are outside the range. The criteria values and the timelines provide detailed context of each similar record and enable users to spot check the results.

3.3.1.4 Similar Record Overview

Criteria value distributions of the similar records are shown at the top of Figure 3.1d to provide an overview of the results. The colors of the bars are consistent with those in the criteria control glyphs, where green bars represent criteria values within the tolerance range, gray bars represent those outside the tolerance range, and the triangles show the value of the seed record. My initial design overlaid the distributions of similar records on the distributions of all archived records (Figure 3.1b) using the same axes. However, the number of similar records is usually very small compared to the entire population, making the bars difficult to see clearly.

The bottom of Figure 3.1d shows the distribution of the distance scores of all archived records (gray bars) and similar records (green bars). The average distance scores are also marked on the chart. This distribution provides an overview about which records are included in the results and how different they are compared to the entire population.

3.3.1.5 Other Configurations

Simpler configurations may be needed to satisfy the needs of intermittent users or to be embedded in specific applications. Advanced users or application designers can configure the visibility of all interface components to provide different levels of controls and context. In the user study, I used three configurations: *Baseline*, *Simple* and *Complex*. *Baseline* provides no controls over the criteria, emulating a black-box interface (Figure 3.3). IDs are only shown to indicate that the search has

completed. *Simple* allows turning on and off each criterion and shows distributions of the results (Figure 3.2).

3.3.2 Search Algorithm

As users add or adjust a similarity criterion, PeerFinder automatically executes the similarity search and updates the results on the display. The search execution consists of two steps. First, a filtering step uses "Exact Match" criteria to eliminate records that do not match. Second, the ranking step uses "Close Match" criteria to sort the records and identify the top most similar records. Details are described below.

3.3.2.1 Filtering

For each criterion marked as "Exact Match" the following process is used: if the tolerance range is not set, only the archived records that have the exact same value (or pattern for temporal criteria) as the seed record will be retained. Otherwise, the records' criteria values need to be within the tolerance ranges to be retained. The tolerance range is represented by a set of values for categorical criteria and by a pair of upper and lower bounds for numerical or temporal criteria.

3.3.2.2 Ranking

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

Figure 3.2: The *Simple* version of PeerFinder provides basic criteria controls (turning on and off each criterion in timeline (a) and record attributes (b)), and simple context (record IDs (c) and overall distribution of the results (d)).

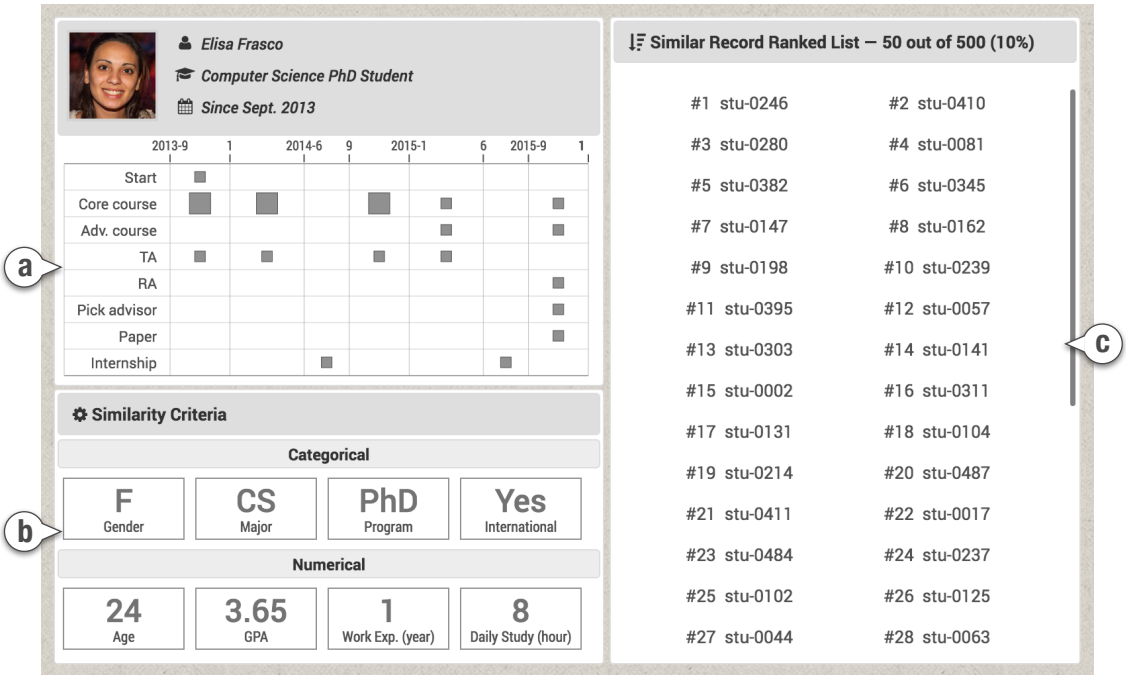

Figure 3.3: The *Baseline* version of PeerFinder provides no controls over the criteria (users can only see the seed record's temporal events (a) and attribute values (b)) and no context (only a list of IDs as results (c)).

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 difference in each criterion and then summarizes them into a single distance score.

Categorical criteria: For each categorical criterion $cc \in C$, I define the difference 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 cc's value of a given record and $t(cc, s)$ returns the set of values in the tolerance range of cc or $\{v(cc, s)\}$ if the tolerance is not specified. I 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 nc's value of a given record and $t_u(nc, s)$ and $t_l(nc, s)$ returns the upper and lower bound of the tolerance range of nc, respectively. When the tolerance of nc is not specified, I 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 $tc \in T$, I compute a value $v(tc, r)$ 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. Since $v(tc, r)$ returns a numerical value, I reuse the difference function for numerical criteria and let $\Delta_T = \Delta_N$.

Finally, I summarize a comprehensive distance score for each pair of archived record r and the seed record s based on weighted Euclidean distance [76]:

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

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

3.4 Evaluation

Searching for similar people to guide life choices is still a new research area and many user studies will be needed to evaluate PeerFinder as it gets embedded in applications that use the ranked list of records to provide guidance. Similarity remains subjective (see early section on challenges) and no ground truth dataset exists, so I chose to focus this first lab study and expert interviews on gaining insights into factors that engage users and promote more trust in the results.

3.4.1 User Study

A within-subject user study compared three versions of PeerFinder (Figure 3.1-3.3) using different levels of complexity (*Baseline*, *Simple*, and *Complex*), as a combination of control and context. The goal was to understand how the levels of controls and context affect users' engagement and their confidence in the ability of the results to be useful. I were interested to see if users would defy conventional guidelines and prefer a more complex interface that demanded more time to use. I also wanted to get feedback to improve the interface.

3.4.1.1 Participants and Apparatus

I recruited 18 university students by email (10 males and 8 females, aged 20–30, $M = 24.67$, $SD = 3.12$). Ten of the participants had technical backgrounds and were experienced in software development, statistics, and data analysis (from the Information School or the Department of Computer Science). The other 8 had limited technical backgrounds but used computers in their study, e.g., web design or print design in the Art Department. None of the participants had prior experience with PeerFinder. Each participant received 10 dollars. A desktop computer was used, with a 24-inch display of resolution 1920×1200 pixels, a mouse, and a keyboard.

3.4.1.2 Datasets for Evaluation

I constructed three synthetic datasets with realistic but simplified features to test the three PeerFinder designs. Each dataset contained 500 records of archived university students. The records had three categorical attributes: gender (male or female), major (Computer Science, HCI, Math, Art), program (B.S., M.S., Ph.D.), and international student (yes or no); four numerical attributes: age (when they started school), GPA, previous work experience (year), and average study time per day (hour). Eight categories of temporal events were included, including "start school", "core course", "advanced course", "paper", "TA (Teaching Assistant)", "RA (Research Assistant)", "pick advisor", and "internship". On average each archived record contained 35 events over 5 years. I generated record attributes with normal and binomial distributions. For temporal events, I reviewed real data and included similar patterns with random variations. The names of events and attributes are generic so that all students can conduct the tasks.

I originally wanted to customize the seed record to match the participant's own data and ask them to search for students like themselves, but I decided against this strategy to normalize the task and avoid privacy and confidentiality issues. Instead, I handpicked a record (named Elisa Frasco and illustrated in Figure 3.3a) that would serve as the seed record: a female international student, majoring in Computer Science and currently in the third year of her Ph.D. study. She is 24 years old and has one year of work experience before starting graduate school. On average, she spends 8 hours on study each day and maintains a relatively high GPA

of 3.65. The timeline showed no papers in the first two years, internships in the last two summers, work as a TA all along except for an RA position in the last semester, after picking an advisor.

3.4.1.3 Hypotheses

My hypotheses were:

H1. Users' confidence will be the highest with *Complex* and the lowest with *Baseline* in that the result set is similar enough to the seed record to provide evidence to guide making academic plans.

H2. Users will prefer *Complex* and *Simple* over *Baseline*.

H3. Users will spend the longest time using *Complex* and the shortest time using *Baseline*.

H4. Users will make more result refinements using *Complex* than *Simple*.

H5. Users will give higher ratings for ease of learning and ease of use for *Simple* and *Baseline* than *Complex*.

I hypothesized that users would spend longer time (**H3**) and make more result refinements (**H4**) in *Complex*, thus increasing their trust in the results (**H1**) and preference for the interface (**H2**). **H3** and **H4** were also an attempt to capture user engagement. Ease of learning and ease of use (**H5**) was included to replicate prior research showing that added complexity reduces ease of learning and ease of use and contrast the results with preferences [77, 78].

3.4.1.4 Procedure

After the initial email recruitment, I sent more detailed directions: *"You will be asked to (1) learn about a (hypothetical) close and important friend of yours who needs advice to improve her academic plan, such as when to take advanced classes, whether to intern during the summer, or when to try to publish papers, and (2) use three different user interfaces to search for students similar to that friend. Data from those similar students will be used as evidence to provide guidance for your friend. You will not be asked to provide or review the guidance itself, only to select a set of similar students."* The record of the hypothetical friend was also provided and participants were encouraged to get familiar with it.

In the lab, each session lasted about 60 minutes. In a brief general training (about 5 minutes), the experimenter made sure that participants were familiar with the task and the hypothetical friend, and answered questions. Next, one of the three versions of PeerFinder (*Baseline*, *Simple*, or *Complex*) was used and the participants were shown a short tutorial (max 5 minutes) covering its interface and operations. The experimenter answered questions and encouraged them to think aloud. The participants were reminded to care about their friend and there was no time limit for the task. When satisfied with the results, the participants needed to click a "finish" button and complete a user satisfaction questionnaire using a 7-point Likert scale:

Q1. How easy was it to learn the interface (1=very difficult, 7=very easy)?

Q2. How easy was it to use the interface (1=very difficult, 7=very easy)?

Q3. How confident were you that the records in the results were similar enough to your friend in order to provide evidence to guide her making academic plans (1=not confident at all, 7=very confident)?

The training, task, and questionnaire were repeated with the other two versions using different datasets so that the results varied. Interface order and datasets were counterbalanced. Participants were allowed to see and adjust the subjective rating they gave for previous versions. Task completion times and numbers of result refinements (i.e., the number of adjustments in criteria controls) were recorded automatically. After using all three versions, participants were asked to rank them based on preference and debriefed to collect feedback.

3.4.1.5 Results

Repeated Measures ANOVA was applied to compare the completion times (log-transformed) and numbers of result refinements, and paired t-test was used for post-hoc comparisons. For questionnaire ratings, I used Friedman test and pairwise Wilcoxon test. All tests used a significance level of 0.01.

Questionnaire: As reported in Figure 3.4, *Baseline* was rated the easiest to learn in Q1 followed by *Simple* and *Complex*. Significant differences were found among the ratings ($\chi^2(2) = 28.00$, $p < 0.001$). Follow-up comparisons indicated that all pairwise differences were significant. The average ratings in Q2 showed the same order of the three versions for the ease of use and the differences were significant ($\chi^2(2) = 32.11$, $p < 0.001$). Pairwise comparisons found significant differences

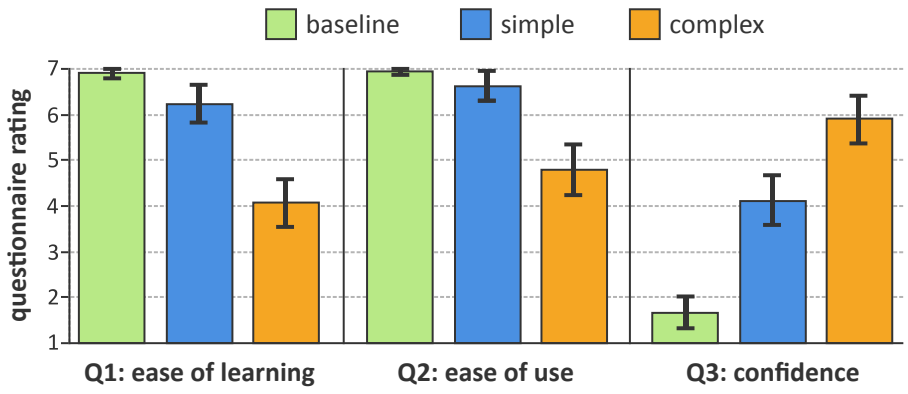

Figure 3.4: Average ratings for each version of PeerFinder in the user satisfaction questionnaire (error bars show 95% confidence intervals). 1=very difficult and 7=very easy in Q1 and Q2; 1=not confident at all and 7=very confident in Q3.

between *Complex* and *Baseline* and between *Complex* and *Simple*. These results supported **H5**.

In Q3, *Complex* had the highest confidence rating ($M = 5.89$) followed by *Simple* ($M = 4.11$) and *Baseline* ($M = 1.67$). Significant differences among the ratings were detected ($\chi^2(2) = 32.14$, $p < 0.001$) and all pairwise differences were significant, which supported **H1**.

Completion time: On average, the participants spent 0.65 minutes ($SD = 0.34$) on *Baseline*, 6.16 minutes ($SD = 2.12$) on *Simple*, and 16.03 minutes ($SD = 6.17$) on *Complex* (Figure 3.5a). Significant differences were found in the log-transformed completion times across these three versions ($F_{2,34} = 248.42$, $p < 0.001$). Post-hoc comparisons showed all pairwise differences were significant, supporting **H3.**

Result refinement: On average, the participants made 16.39 refinements ($SD = 14.08$) using *Simple* and 34.17 refinements ($SD = 16.90$) using *Complex* (Figure 3.5b), which was a significant increase of 108%, supporting **H4**.

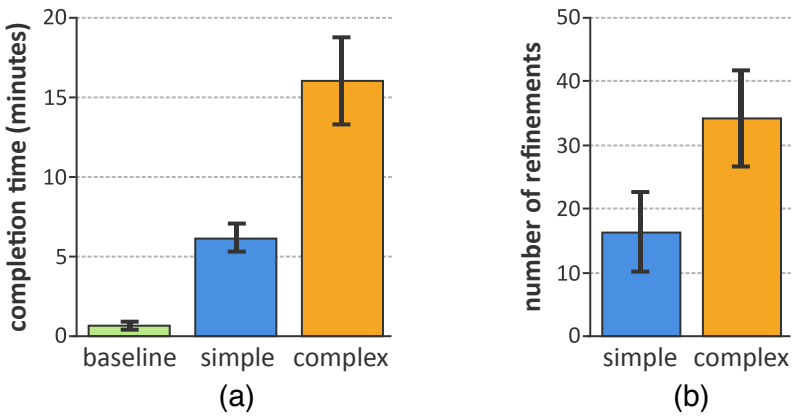

Figure 3.5: (a) Average completion times and (b) average numbers of result refinements using different versions of PeerFinder (error bars show 95% confidence intervals).

3.4.1.6 Preference and Feedback

16 out of 18 participants chose *Complex* as their preferred interface. Two picked *Simple*, and *Baseline* was always the least favorite, which confirmed **H2**.

Ease of learning and use: Although *Baseline* was rated as the easiest version to learn and to use, many participants commented on their disappointment, e.g., *"I can do nothing."* 9 participants commented that *Simple* offered a good balance of simplicity and capability, e.g., *"I like the binary controls and clear presentation of the results. I felt more focused."* Another who preferred *Simple* explained that *"the controls satisfy my needs and the Simple interface is easier to explain to the friend I am helping."* As for *Complex*, 11 participants gave a neutral rating in Q1 or Q2 and 4 of them suggested that *"it requires training and practice to become familiar with this interface."* In contrast, one participant who thought *Complex* was easy to learn explained: *"The graphs are the same everywhere. After understanding one, I understand others."*

Confidence: All participants expressed lacking trust in the results generated by *Baseline* and the most common feedback was *"the results look random."* One participant emphasized: *"I want to know how the algorithm gives these IDs."* Another added: *"The results may be good but without any details, I am skeptical about it."* 15 participants gave higher confidence ratings for *Complex* than *Simple*. The most common reason given was control: *"The advanced controls enable me to get more precise results,"* or *"when using the Simple version I can see some flaws in the results but cannot fix them,"* or *"since I have the functionalities to do more, I am more motivated to pay attention and try different settings."* Participants also appreciated seeing the similar records provided by *Complex*: *"it helps me verify the results and correct small mistakes"* or *"seeing concrete students provide inspirations for tuning the controls and specifying temporal patterns."*

Participants expressed concerns about the complexity of the *Complex* version. One described: *"There are many options and data you need to keep track of. It was like piloting a plane."* Another said that *"the [similar] student information were distracting when I was not using it."* One participant who preferred *Simple* commented: *"My trust diminished every time I got lost. I worried about missing anything."*

Search strategies: Most users only briefly reviewed the display of *Baseline*. On the other hand, I observed users repeatedly turning on and off criteria in *Simple* and inspecting the result distributions to see the effects. When using *Complex*, users commonly carefully reviewed the criteria one by one and tried different settings. They kept an eye on the result distributions, and reviewed the details of a few

similar records to verify their settings. At the end, many scanned the entire list of similar records looking for problems.

Suggestions: Suggestions included starting simple and allowing users to add controls and details as needed, enabling users to choose colors and interface layout, marking important records. Automatic aids were also requested: recommending criteria settings to save users' effort and detecting outliers in the results for users to review. Usability suggestions included making buttons more noticeable, flipping the layout entirely to show the seed at the top and results below, and merging both distributions (for the population and the seed record) into one.

3.4.2 Expert Review

I conducted one-on-one 45-minute interviews with domain experts whose professional activities involved providing guidance to others: three student advisors (E1-3) and a physician (E4), each having at least 10 years of experiences. I demonstrated the three interfaces using the same datasets as in the user study and asked the experts to explore on their own. I answered questions and recorded comments and preference.

All four experts expressed great enthusiasm for PeerFinder: *"it helps me provide advice based on data and avoid false assumptions"* (E1), *"it provides a new method to make use of the collected student data"* (E2), and *"it provides a faster and data-driven way to quickly profile a student and start the conversation"* (E3). E1 and E3 preferred *Complex*. E1 suggested allowing users to re-arrange and turn

on and off each view since different views are used at different stages during the exploration. E3 wanted to look at the "future" activities and outcomes of those similar students. E2 picked *Simple* stating that *"the interface is simpler and helps me communicate the results with other advisors or students."* E4 stated that all three versions have values depending on usage: *"The Complex version could be very useful for patients working on their own for health maintenance. For regular doctor visits the Simple version may work better since the time is very limited."* He mentioned that diseases usually have their own schema which can be used as presets for the criteria settings. The importance of privacy protection was repeated, and ethical issues were mentioned, for example: *"Some students may be demoralized by the worst cases in the results (similar records)."*

3.5 Discussion

All hypotheses were confirmed with size effects larger than I expected. For example, I expected more participants would prefer the *Simple* version, but despite the increased complexity, the *Complex* version was preferred by the majority of users. Engagement, as measured by time spent on task and number of interactions, was also higher when using *Complex*. More importantly, confidence was higher when using *Complex*. These findings suggest that users should be provided with controls over the search process when making life choices. The lab study was tied to a particular scenario (student advising) but my research emphasizes that different situations of use require different criteria to be used, reinforcing the importance of

customization for the end user and for the application developer. While some of the challenges remain (e.g., no ground truth), I believe that there is value in clarifying those challenges, and that the PeerFinder prototype and evaluation approaches (e.g., measuring trust) will inspire others to develop better solutions to these challenges.

Reviewing ethical issues is important. 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 [79]. PeerFinder mitigates this issue by giving transparent data access to both advisors and advisees and involving them in the decision-making process.

My user study had several limitations. I tested only three configurations, omitting alternate versions, for example, one that included no control but provided rich context. Testing all nine configurations will help tease out the separate roles of increased control and increased context. I chose a within-subject design so the *Baseline* may have seemed more disappointing to participants who saw other versions first. Between-group studies may affect the differences in confidence, but then preference cannot be collected. In my study, I made sure that there were records similar to the seed record, but even with "big data" there may be cases where few similar records exist. In those cases, we need to verify that user confidence remains low. I

did not evaluate the accuracy of the search algorithm because ground truth is not available. I hope that increased interest in this topic will lead to the development of benchmark datasets. In the meantime, the search algorithm can be improved to handle multi-attribute data, treat ordinal attributes separately, and incorporate refined similarity measures for temporal patterns. Lastly, my study focused on a student advising scenario. Medical scenarios are likely to be more complex unless the tool is customized to a carefully chosen medical specialty and diagnosis. In the future, I also hope to incorporate outcome data and help users identify the similarity criteria that are most correlated to the outcomes of interest.

3.6 Summary

People often seek examples of similar individuals to guide their own life choices. This chapter characterized the challenges facing designers and evaluators of systems supporting this task. It described PeerFinder, a prototype interface that enables users to interactively find and review records based on similarity to a seed record using both record attributes and temporal event information. While there is still much to do to improve the interface, my user study with 18 participants suggested that users are more engaged and more confident about the results when provided with more control and more context, even at the cost of added complexity. The following chapters present the design process of advanced visual interfaces for finding similar and dissimilar people and report on case studies of PeerFinder embedded in real applications.

Chapter 4: Advanced Visual Interfaces for Finding Similar and Dissimilar People

Recommendation applications can guide users in making critical life choices by referring to the activities of similar peers. With the rapid accumulation and digitization of personal records, software tools have been developed to enable the retrieval and analysis of the data of similar individuals to facilitate making critical decisions. For example, patients and their physicians may explore data from similar patients to select the best treatment (e.g., PatientsLikeMe [39], CureTogether.com). Students making academic plans may be inspired by the achievements of similar students (e.g., PeerFinder [70], EventAction [67]). While automated black-box recommendation techniques are effective and used widely in shopping and entertainment applications [18, 19, 80], transparency is critical when users review data and recommendations for life decisions, carefully decide to accept a recommendation, or remain doubtful [31, 71]. In this chapter, I focus on how to improve the selection of peer groups, i.e., how to select "people like me," or "people like the patient, student, or customer I am advising."

Previous work suggested that users are more engaged and more confident about making critical life choices when provided with more controls and more context, even

at the cost of increased complexity [70]. The next question then becomes: which controls and context should be provided? How do users find a satisfying peer group? And how can I facilitate this process?

In this chapter, I report on three visualization designs and three analytic workflows to support users in retrieving, reviewing, and refining peer groups, making use of both record attributes and simple patterns of temporal events found in the record[1]. I introduce LikeMeDonuts, a novel hierarchical visualization providing an aggregated overview while preserving details about individual peers, to support users in reviewing similarities and differences of a group of records compared to the seed record. While most existing tools focus on hierarchies that have a fixed structure (e.g., the ICD-10 codes [74] or phenotypes [82]), I investigate situations when the order of the hierarchy is flexible and subjective, depending on the analysis goals and users' preferences. My prototype provides controls for users to interactively adjust the layout, create visual representations that best satisfy their needs, and refine the peer group composition. It also provides recommendations on improving the layout so as to reduce visual clutter and mitigate issues of scalability.

I refined the design through three rounds of formative usability evaluation with a total of 12 target users, and report how the prototype evolved on users' feedback. I propose three analytic workflows for forming peer groups and report on users' experience and preferences.

My contributions include:

- A novel hierarchical visualization (LikeMeDonuts) that provides an overview

[1]This work was published at ACM TIST 2018 [81].

of peer groups with a flexible hierarchy of criteria values, similarity encoding, and interactive support for trimming the peer group.

- An interactive visualization system (iteratively refined through three rounds of formative usability study) that combines three new visualization components and supports three analytic workflows.

4.1 User Interface

After describing motivation and goals, this section describes the final design of the interface. The rest of the chapter will describe early designs, problems uncovered during three rounds of usability testing, and how the design evolved. Finally, the discussion section addresses remaining challenges and possible solutions.

4.1.1 Motivations and Needs Analysis

In PeerFinder [70], I described user studies investigating how the complexity of the interface affects users' engagement in the decision making process and confidence in the results. I used two visualization components, barcharts and a ranked list, and evaluated the interface through a user study with 18 university students and interviews with 4 domain experts (three student advisors and a physician). Based on my discussions with the participants, I identified two critical users' needs which motivate the design of the new interface components introduced in this work:

N1. Tracking across multiple criteria. The interface should allow users to track and review a group of records that share similar values across multiple

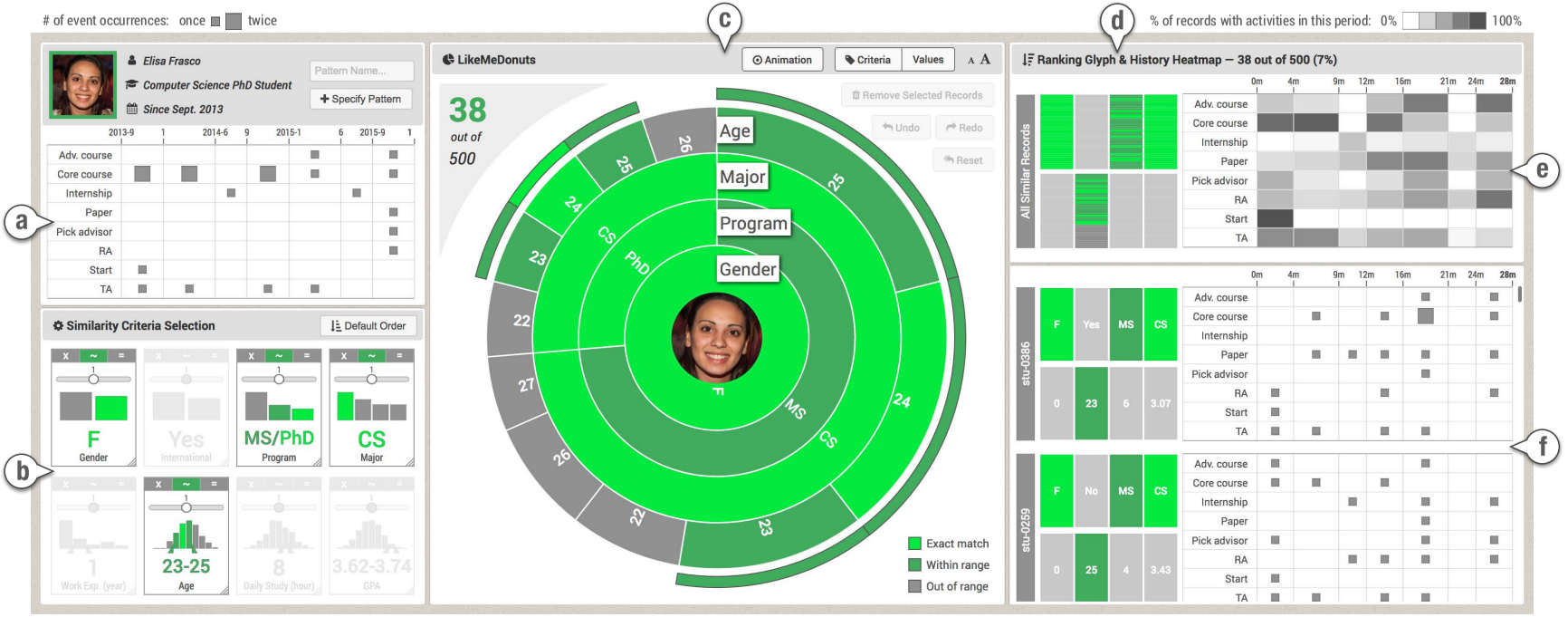

Figure 4.1: The interface of my prototype for forming peer groups: (a) seed record timeline, (b) similarity criteria controls, (c) LikeMeDonuts representing criteria values of the 38 most similar records as a hierarchical tree, (d) Ranking Glyph providing a compact overview of 38 most similar records ranked by similarity, (e) History Heatmap showing the popularity of the temporal events among similar records, and (f) ranked list of similar records, displaying detailed individual information.

criteria, so that users can estimate the size of the group, explore how those records are distributed in other criteria, and refine the results by removing the group when necessary. The barcharts in my original design only support showing the value distribution of each separate criterion.

N2. Reviewing results at different levels-of-detail. The interface should provide both individual-level details and group-level overviews so that users can efficiently review and refine the results of similar records using both record attributes and temporal events. While the ranked list in my original design was useful to display full details of individual records, users were unable to get an overview of those records.

To satisfy these needs, I designed three new visualization components for reviewing and refining peer groups. My main goal when designing LikeMeDonuts (Figure 4.1c) was to reveal distributions across combinations of multiple similarity criteria (e.g., female students majoring in Computer Science and having GPAs higher than 3.5). LikeMeDonuts allows users to estimate the size of multiple groups of records (i.e., the branches in a hierarchy of criteria) and provides interactive controls for selecting or removing groups, and rearranging the hierarchy that shapes those groups (**N1**).

The purpose of Ranking Glyph (Figure 4.1d) and History Heatmap (Figure 4.1e) was to provide a compact overview of the ranked list of the similar records (**N2**). The Ranking Glyph aimed to help users understand how similarities and differences for each criterion evolve as they go down the ranked list of similar records

(e.g., are students having two internships more likely to be ranked on the top?). The History Heatmap helps users inspect common temporal patterns of activities for the entire peer group—or a selected subset (e.g., are students like me still taking classes in the fourth year?).

Those new components are integrated into the existing PeerFinder interface, which provides basic interface components: the seed record timeline (Figure 4.1a), similarity criteria controls (Figure 4.1b) and the underlying similarity search algorithm, and the basic ranked list of similar records for displaying detailed information (Figure 4.1f). Those basic components have also been refined as a beneficial side effect of the usability study (e.g., consistent use of color and improved coordination between components). In the rest of this section, I describe the basic interface components first, then I present the new components in greater detail.

4.1.2 Basic Interface Components

Seed record timeline. The seed record's history of activities is shown as an aggregated timeline in a timetable (Figure 4.2a), 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 4.2, two temporal patterns have been specified based on the seed record's internship (having an internship every summer) and research activities (no papers

Figure 4.2: Four of the basic components that refine the PeerFinder interface: (a) seed record timeline, (b) similarity criteria controls, (c) similarity distribution, and (d) similar record distribution. In this example, a total of 10 similarity criteria are used, including two temporal criteria in the bottom row. The mouse cursor is hovering on the temporal criterion of "no papers in the first two years and late selection of an advisor." This criterion and the corresponding temporal pattern are highlighted in orange.

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 4.2b) showing its name, the value for the seed record and the distribution for all archived records. Users can select how the criterion is to be used: "Ignore" ($\times$), allow "Close Match" ($\sim$), or require "Exact Match" ($=$). A tolerance range can also be defined to treat multiple categorical values or a range of numerical values as 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 (see PeerFinder [70] for algorithmic details) and a histogram of the scores is displayed (Figure 4.2c). Users can adjust the portion of the histogram that is selected for the results, i.e., the peer group. In Figure 4.2c, the top 20% most similar records (100 out of 500) are selected. Since the similarity scores change when users adjust the criteria controls, I 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 4.2d). 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.

Basic 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 4.1f). For privacy, the individual records will need to be hidden when users do not have proper viewing permission [70]. Part of the overviews or their labels may also need to be hidden when the number of records included is too low.

Improvements have been made to the basic interface components, e.g., the new color scheme used in the LikeMeDonuts was propagated to older components, and brushing and linking capabilities were added to coordinate all the views.

I now describe the new visualization components.

4.1.3 LikeMeDonuts

LikeMeDonuts is a radial space-filling visualization that shows the criteria values of the similar records as a hierarchical tree (Figure 4.3). 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

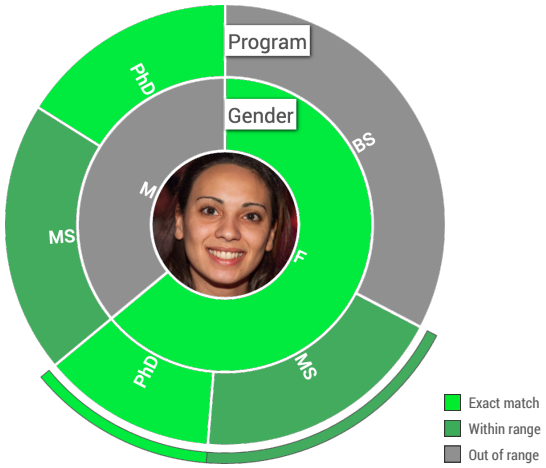

Figure 4.3: This LikeMeDonuts shows two criteria as a two-level hierarchical tree. An image of the seed record is placed at the center. The inner ring represents gender. It shows that most records in the peer group are females like the seed record. The males are shown in gray, indicating that they are outside the tolerance range. The outer ring is for program. Among the females, most are B.S. students, and some are M.S. (shown in dark green because they are within range but not exactly like the seed record) or Ph.D. students. The males are all M.S. or Ph.D. students. The thin partial ring outside the donuts highlights records that are within range for both criteria.

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 4.4, I use the empty corner space to display contextual information and controls. The top left shows the number of similar records being

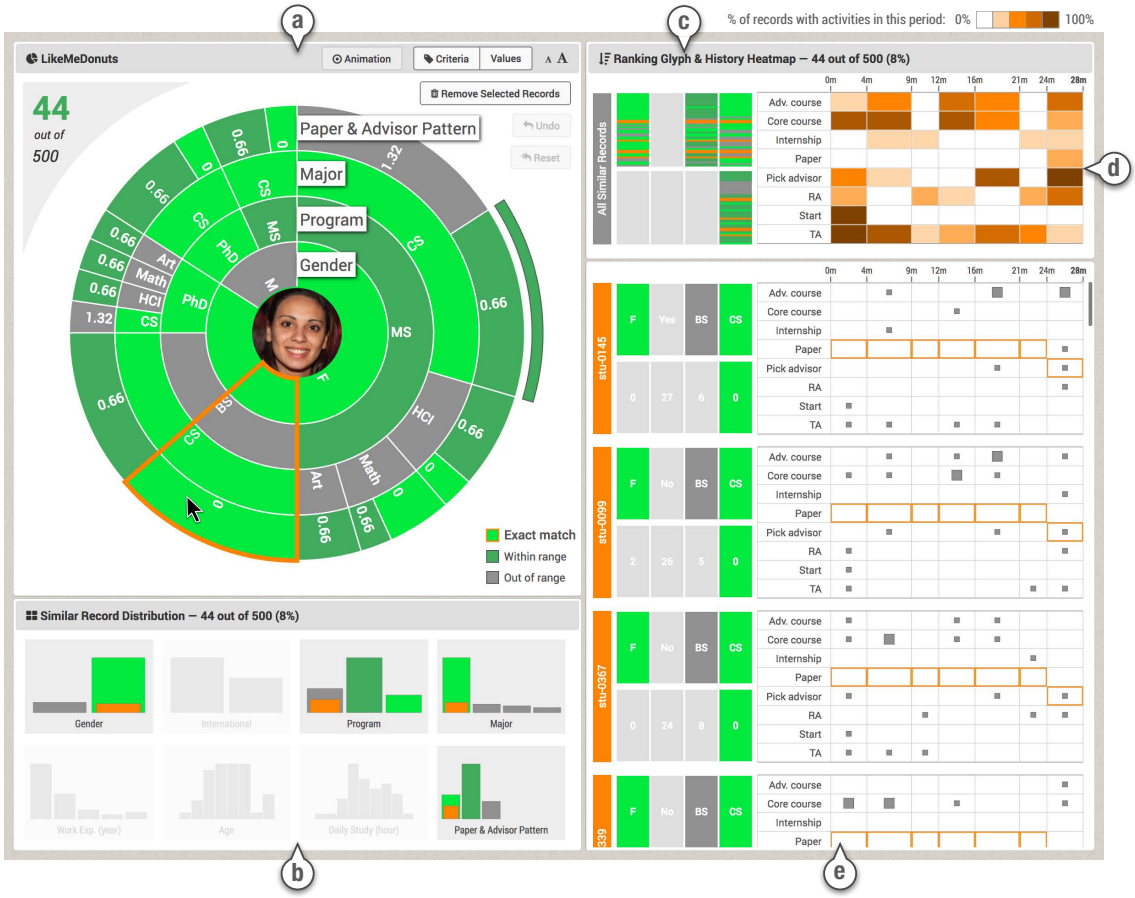

Figure 4.4: All views are coordinated. In this example, a group of records are selected in the LikeMeDonuts (a) and therefore highlighted in orange in the similar record distribution (b), the Ranking Glyph (c) and the selected records are brought to the top of the similar record ranked list (e). The History Heatmap (d) is also updated to show only the events from the selected records. A "Paper and Advisor" temporal pattern was included in the criteria and appears as a numerical distance score in the LikeMeDonuts (with smaller values indicate more similar). The location of the pattern is also highlighted in the timelines of the individual records.

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.

4.1.3.1 Interactions

The donut rings and ring sectors are responsive to users' interactions and are linked to other visualizations on the interface. Hovering on a criterion in the similarity criteria controls highlights the matching donut ring with an orange border. Hovering on a ring sector highlights records represented by that sector with orange borders. When users click on one or multiple ring sectors, the selected records are highlighted in other visualizations (Figure 4.4): (1) orange bars are added in the similar record distribution barcharts, (2) the ranking of the selected records is shown in orange in the Ranking Glyph, (3) the History Heatmap shows the temporal activities of the selected records—using a color gradient from dark orange to white, (4) the individual selected records are be moved to the top of the ranked list of records with their IDs colored in orange, and (5) if a temporal criterion is used, the patterns will be highlighted with orange borders in the timelines of the similar records.

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.

4.1.3.2 Animated Transitions

I carefully designed a four-stage animation [83,84] to clarify the transition that occurs when users adjust the criteria controls or edit the peer group at the record level. The first stage fades out records removed from the peer group and criteria set to "Ignore" (i.e., removed). In the second stage, the LikeMeDonuts is resized to fill the screen space made available by removed donut rings or make space for new donut rings that will need to be added later. The third stage adjusts the size and color of the ring sectors and reorders them according to the updated peer group. The last stage fades in those newly added records and criteria/rings. A stage will be skipped if no changes occur during it. Each stage is set to 500 milliseconds. The entire animation takes two seconds at most for adjusting criteria controls, and one second for making an edit at record level (only involving the first and third stages). Users can turn the animation on or off.

4.1.3.3 Order of Donut Rings

Given a set $\mathbf{C}$ of n criteria, the number of donut ring sectors is:

$$number\ of\ sectors = \sum_{i=1}^{n} \left(\prod_{j=1}^{i} \|c_j\| \right) \qquad c \in \mathbf{C}$$

where $\|c\|$ is the number of unique values of a criterion and as j increases, c_j moves from an inner ring to an outer ring. Note that $\|c_j\|$ appears in $(n - j + 1)$ terms of the summation. Therefore, inner rings have a larger impact on the result than outer rings. To minimize the number of sectors, criteria with smaller numbers of possible

values should stay in the inner rings, whereas those with larger numbers of possible values need to be placed in the outer rings. My system recommends an order of the donut rings at the start that minimizes the total number of sectors (therefore setting the default order of criteria in all other views). Users can then rearrange the rings to create views that better match their preferences by dragging the rings inward or outward, or dragging the criteria glyphs in the criteria control panel (Figure 4.1b).

In summary, the LikeMeDonuts is a novel and highly customizable overview of a peer group that allows users to rapidly evaluate the similarities and differences of records in the group compared to the seed record. Interaction allows users to remove subsets directly in the LikeMeDonuts, spot matching controls or records in other coordinated views, and reorganize the rings.

4.1.3.4 Alternative Designs

Before settling on a sunburst-like circular layout, I explored alternative designs for presenting the similar records and the similarity criteria. I tested parallel coordinates [85] and radar plots [86], two common designs for visualizing multi-dimensional data. They were effective at revealing patterns between adjacent dimensions. However, since the dimensions are not hierarchically structured, it is difficult to track a group of records that share similar values across multiple criteria (e.g., male patients aged around 60 with Hyperglycemia) or to show the size of a group. Also, parallel coordinates have severe overlapping issues when displaying categorical values.

I also tested icicle plots and Treemaps [87], but as I compared all those designs

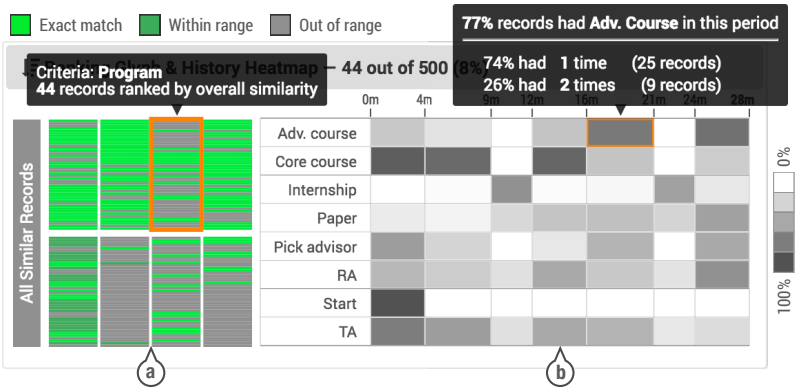

Figure 4.5: (a) Ranking Glyph and (b) History Heatmap summarizing both criteria values and temporal activities of 44 most similar records. The figure includes two separate tooltips that would be shown when hovering on a glyph or a time period of the heatmap. In the Ranking Glyph, I see that the top portion of the highlighted "Program" glyph has few green bars. In comparison, for the "Paper & Advisor Pattern" glyph (second row, fourth column) most green matching records are at the top, indicating that the top records have the right pattern and that this criterion may have a strong influence on the overall similarity.

my desire to center the design around the seed record (and a photo of the person) become stronger and I narrowed my design space to only circular designs. The classic sunburst design was enhanced and adapted to my application: (1) the hierarchy of similarity criteria can be reordered, (2) a set of operations allow users to modify the hierarchy and layout based on preferences, and (3) the photo at the center provides a visual reminder that all the information is relative to that person.

4.1.4 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 4.5a). 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 4.1b) and the record ranked list (Figure 4.1f). 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.

4.1.5 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 4.5b). 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 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.

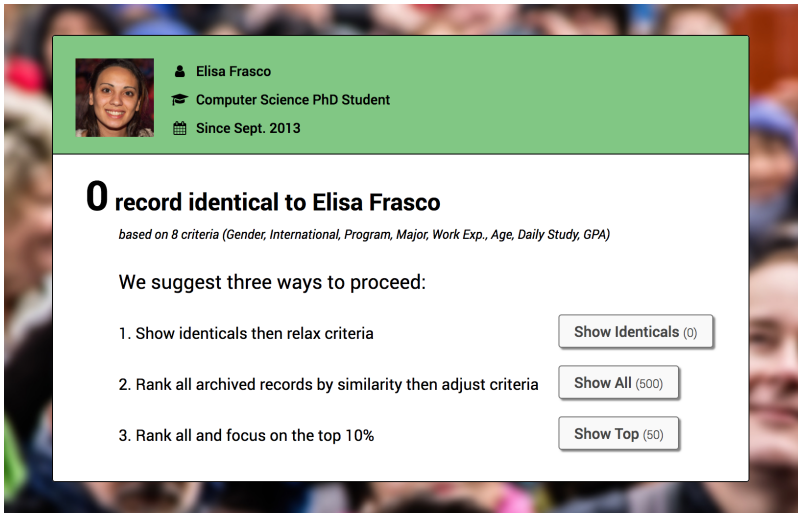

Figure 4.6: The startup screen that shows basic information of the seed record and suggests three workflows for users to start the analysis: (1) show identical records, (2) show all archived records, and (3) show top 10% most similar records.

4.1.6 Support for Analytic Workflows

The first thing users typically do is to select the seed record. This would be done in the larger application in which PeerFinder may be embedded (e.g., EventAction [67]). Then the startup screen displays basic information about the seed record and provides a choice of three possible ways to proceed, i.e., three workflows[2], which come with explanations (Figure 4.6). These three workflows were born from the observations and interviews of users of the initial version of PeerFinder [70], and from my own discussion of possible ways to get started with the process.

The "Show Identicals" workflow helps users start with a small set of records and then relax constraints. It presents users with all similarity criteria set to "Exact Match" at the start and finds only identical records. Users can then adjust tolerance

[2]A demo video illustrating the interface and workflows is available at `http://hcil.umd.edu/peerfinder`.

ranges or relax some of the criteria to "Close Match" to find a larger set of records with similar values. The second workflow is "Show All", which starts by selecting everybody and letting users review how the seed record differs from the entire population. All criteria are set by default to "Close Match" and all records are selected in the similarity distribution panel (Figure 4.2c). Users can then narrow the results by switching the criteria to "Exact Match", narrowing tolerance ranges and thus reducing the total number of records in the results. The "Show Top" workflow also uses "Close Match" for all criteria at the start but narrows the results to the top 10% most similar records. Users can further adjust the criteria and the similarity range to narrow or expand the results.

4.1.7 Interface Configuration Panel

On the top of the interface is a configuration panel that allows users to control the visibility of each interface component, so that different interface configurations can be used during different analysis stages. Pressing the "ESC" key will show and hide the interface configuration panel. Users can also rearrange the layout by drag-and-drop of interface components. In the usability study, I saw participants hide the criteria control panel and similarity distribution panel after they were happy with the criteria settings, and move the History Heatmap next to the seed record timeline to compare the activity patterns.

4.2 User Study and Iterative Design Process

The new design evolved over three rounds of a formative usability study to evaluate the comprehensibility and learnability of the interface components and gain an understanding of users' analytic workflows in forming groups of similar people. I summarize the study procedure and report on users' feedback, describing how my prototype evolved.

4.2.1 Participants and Apparatus

I recruited a total of 12 university students by email (5 males and 7 females, aged 22–31, $M = 26.33$, $SD = 3.08$). All participants used computers in their study. The entire study was spread over three rounds during a month. In each round, I conducted study sessions with four participants and iteratively improved the prototype based on their feedback. A desktop computer was used, with a 24-inch display of resolution 1920×1200 pixels, a mouse, and a keyboard. Each participant received 10 dollars.

4.2.2 Dataset

I constructed a synthetic dataset of 500 archived records of university students with realistic but simplified features. The records had four categorical attributes: gender (male or female), major (Computer Science, HCI, Math, or Art), program (B.S., M.S., or Ph.D.), and international student (yes or no); four numerical attributes: age (when they started school), GPA, previous work experience (year),

and average study time per day (hour). Eight categories of temporal events were included: "start school", "core course", "advanced course", "paper", "TA (teaching assistant)", "RA (research assistant)", "pick advisor", and "internship". On average, each archived record contained 35 events over 5 years. I generated record attributes with normal and binomial distributions. For temporal events, I reviewed real data and included similar patterns with random variations. The names of events and attributes are generic so that all students can conduct the tasks.

I handpicked one of the synthetic records to serve as the seed record. Her name is Elisa Frasco. The photo is authorized for using in mock-ups[3]. Elisa is an imaginary female international student, majoring in Computer Science and currently in the third year of her Ph.D. study. She is 24 years old and has one year of work experience before starting graduate school. On average, she spends 8 hours on study each day and maintains a GPA of 3.65. Her timeline shows no papers in the first two years, internships in the last two summers, working as a TA all along except for an RA position in the last semester, after picking an advisor.

4.2.3 Procedure

Each session lasted about an hour. During the first five minutes, the experimenter made sure that participants were familiar with the task and the hypothetical friend. I told participants: *"You will be asked to (1) learn about a (hypothetical) close and important friend of yours who needs advice to improve her academic plan, such as when to take advanced classes, whether to intern during the summer, or*

[3] `https://randomuser.me`

when to try to publish papers, and (2) use a user interface to search for students similar to that friend. Data from those similar students will be used as evidence to provide guidance for your friend. You will not be asked to provide or review the guidance itself, only to select a set of similar students." The record attributes and temporal events of the hypothetical friend were provided in a table and participants were encouraged to get familiar with it. Questions were answered.

Next, the startup screen of the interface was shown (Figure 4.6), and participants were encouraged to think aloud, explain the decisions they made, and comment on the interface. Participants decided what workflow option they wanted to use on their own and entered the main interface. No training was provided prior to the start. Participants explored the interface on their own and used the similarity criteria controls and visualizations to complete the task. If a participant was stuck for three minutes not being able to do what they wanted (e.g., did not understand an element of the interface) the experimenter provided hints and answered questions. The participants were reminded to care about their friend and there was no time limit for the task. The study session ended when the participant was satisfied with the peer group. If they had not used a component of the interface, the experimenter asked them to try it. At the end of the session, I asked participants to go back to the startup screen and try the other workflows. I collected learnability problems, comments, and suggestions for improvements.

4.2.4 Results and Evolution of the Design

I report on the participants' preferences toward the three workflows, and then focus on the three new visualizations. I report on users' feedback and describe how my prototype evolved.

4.2.4.1 Analytic Workflows

All participants seemed able to understand the workflow options provided on the startup screen on their own. The "Show Identicals" workflow was the most popular and was selected by seven out of 12 participants. "Show Top" and "Show All" workflows were used by three and two participants, respectively. Two participants in the first round complained that it was hard to anticipate the amount of data they would have to look at using the three options. I addressed this issue by adding the number of records next to the workflow options. After completing the task, I asked participants to try the other workflow options for 5 minutes each and rank the three options by preference. Three participants who had initially selected "Show Identicals" during the task changed their mind. Eventually, "Show Top" was the favorite of 5 participants, followed by "Show Identicals" (4) and "Show All" (3). Since there was no clear winner, I decided to keep all three workflows. In the future, usage logs from a larger number of users might help identify an adequate default workflow.

One common reason provided for favoring "Show All" and "Show Top" was wanting a larger number of similar records to get started. Comments included: *"it*

shows me a big picture" and *"the overview helps me understand what I am dealing with."* In particular, one who preferred "Show Top" explained *"it starts with a good set of similar records and saves my time,"* and another said *"it guarantees some good results."* The participants who did not like "Show All" complained that showing all the data was overwhelming and one said *"I am lost. Seeing everything equals to seeing nothing."* Another participant pointed out that *"the show all (workflow) is not scalable. It will destroy the visualizations and slow down the system."* Two participants were concerned about the biases in "Show Top" and explained: *"I want to see all the data instead of a small sample picked by the system."*

From the five participants who liked "Show Identicals" I heard comments such as: *"I preferred to start simple"* or *"the (ranking) algorithm was not involved and I had a better feeling of control."* However, others complained that *"it takes a longer time to get enough results"* and that *"start from blank was frustrating. I thought the system was broken."* Two participants pointed out that they would choose "Show Identicals" or "Show All," depending on the analysis, as one said: *"If I have a strong purpose such as predicting my job after graduation, I will start with only identicals and prepare queries based on my questions. Otherwise, I will start with all the data and try different (criteria) settings in a data-driven way."*

4.2.4.2 LikeMeDonuts

All participants were able to understand the meaning of the donut rings on their own. The color scheme was also understandable. One participant applauded

that *"the color scheme is the same everywhere in the system. I learned it from the criteria controls."* Another said *"the color legend and the text labels made it clear to me."* Participants heavily used LikeMeDonuts just after they finished selecting the initial criteria settings. They mainly focused on reviewing the gray sectors and often went back to adjust the criteria controls to *"exclude unexpected records."* All participants left some gray sectors in the final results. One explained that *"I am aware about the gray areas but those criteria are less important. I will filter them out if I want fewer records in the results."* Another participant who deliberately balanced the gender of the peer group said: *"The gray records are not errors but expected. I kept the male students in gray to show the diversity."*

All participants commented positively about the four-stage animated transitions of LikeMeDonuts and everyone mentioned that the animations helped them keep track of the changes. I asked the participants to turn off the animations and explore for a few minutes. One participant was immediately confused and said aloud: *"Already? It updates too fast and I did not even notice."* Another pointed out that *"this is a complex interface. The animation helps me manage it."* However, later in the analysis, five participants changed their mind. One explained *"the animations take time to play and slow down my operations."* Another added: *"As I become familiar with and trust the system, I may want to turn it off."* So providing the option to turn off the animation is required. At the end, all participants strongly agreed that animations are important for new users to learn the system, confirming previous findings (e.g., [88]). Seven participants stated that they will keep the animation active all the time. One said *"it does not take that much time"* and another

emphasized *"I make mistakes sometimes. It helps me verify my operations."*

As for sorting the donut rings, nine out of 12 participants moved important criteria to the inner rings and kept less important ones in the outer rings. One explained that *"I read the donuts from inside to outside"* and another said *"I prefer to keep important things around my friend."* Two participants used the opposite order because *"the outer rings have more space for important criteria."* The last participant used a mixed strategy. He first sorted the criteria by the number of unique values and then by their importance. During the first round of the usability study, no participant had discovered that they could reorder the donut rings by dragging the criteria icons. I improved this by adding a dragging handler to the icons and changed the mouse cursor to a "Move" style when hovering on the icon. This helped all remaining participants discover the feature.

In the first two rounds of the study, three out of 8 participants had not been able to understand the ring sectors on their own. Two blamed the grouping of sectors in the inner rings: *"I see only a few divisions in the inner rings but many in the outer rings. I did not realize that the rings are aligned to show individual records."* They suggested two ideas to improve learnability: (1) removing the grouping and drawing borders to separate individual records, and (2) highlighting individual records with borders when hovering on a sector. I implemented the second solution and kept the grouping, which helps LikeMeDonuts scale to larger numbers of records. The four remaining participants discovered the meaning of the donuts on their own and commented that the highlighting was helpful for reviewing individual records.

The thin partial ring sectors outside the donuts were not available during the

first two rounds of the study. I observed four out of 8 participants pointing fingers at the screen trying to identify individual records with all criteria values in green. One of them explained that *"I wanted to see if there were any identical records after I changed the criteria."* I then designed the thin ring to highlight identical records and all four participants in the third round were able to understand it on their own. *"I found the green arc when I was focusing on a very similar record,"* one commented, *"I immediately understood."*

4.2.4.3 Ranking Glyph

The initial design of the Ranking Glyph came from brainstorming ideas to represent how the top records differed from the bottom records. In the initial prototype, I had placed the Ranking Glyph on the left side, below the criteria control panel. I hoped that users would understand the Ranking Glyph layout from the layout of the criteria. However, none of the participants of the first round of testing guessed the meaning of the glyph. After being explained how the Ranking Glyph worked one participant said that *"it (the glyph) looks like a compressed version of the records"* and that *"they are both sorted by similarity,"* suggesting that the Ranking Glyph should be moved next to the similar record ranked list (Figure 4.1f). I moved the Ranking Glyph to the top of the record list and I also moved the History Heatmap next to it (previously displayed as the background colors of the seed record timeline). These two visualizations combined provide a true overview of the results.

In the subsequent two rounds of usability testing, all participants were able to

understand the Ranking Glyph on their own. The most common learning strategy was to look at individual records first in the ranked list. *"It looks like a barcode of the record list"* one participant commented. Further testing should verify that the glyph is still learnable when the individual records are hidden for privacy reasons.

The participants typically used Ranking Glyph to determine a similarity threshold, as one said: *"I used the barcharts to filter by value and used the glyphs to filter by similarity."* Five participants particularly liked the way the glyphs are sorted. One said *"it is useful for checking how the top 5% and bottom 5% records look like."* Another (who had some data mining background) commented: *"The glyphs can tell me how each criterion influences the overall similarity. I can easily see the trivial (less influential) ones and put less weight on them."*

The main complaint about the Ranking Glyph is its small size. One participant complained that *"it is too small and hard to track individuals. I can see the top 5% records but cannot see the fifth record."* Another said *"I am more willing to interact with the donuts than the ranking glyph. The pixels (bars) are too small."* To mitigate this issue, I connected Ranking Glyph with LikeMeDonuts so that users benefit from both the interactivity of the donuts and the sorted overview of the Ranking Glyph. Specifically, when a subset of records are selected in the donuts, the horizontal bars representing those records are highlighted, showing their rankings in the entire peer group (Figure 4.4c).

4.2.4.4 History Heatmap

In the first round of the study, the history heatmap was on the left side, combined with the seed record timeline (displayed as background color of the seed record event squares in each cell). The first-round participants were not able to tell if it was showing the activities of all archived records or similar records. Therefore, I moved the History Heatmap to the top of the similar record ranked list, right next to the Ranking Glyph. All remaining participants were able to guess the meaning of the color darkness in the History Heatmap on their own. One participant stated: *"The heatmap is intuitive, just like you add up the gray squares in the timelines below."*

Six participants reported findings from the History Heatmap. For example, *"I was able to see the transition from core course to advanced course and hotspots of internships in the summer,"* one said, *"some interesting patterns just jump into my eyes."* I also observed two participants using the History Heatmap to help understand the activity of the seed record, as one explained *"I wonder if my friend has done any abnormal thing."* To better support this task, I now allow users to change the layout of the interface components so the History Heatmap can be moved closer to the seed record timeline to compare the activities side-by-side. In the future, adding a way to compute the difference between two records or between the seed record and the peer group average may be useful for the timeline views.

4.2.4.5 Similar Record Barcharts

All participants were able to understand the similar record distribution barcharts (Figure 4.4b) immediately and could correctly tell the meaning of colors, horizontal axis, and heights. During the analysis, the participants typically used the barcharts to briefly review the criteria distributions when adjusting the criteria controls. One participant explained that *"it helps me verify my settings"* and another added that *"it just looks simpler than other visualizations."*

When asked to compare barcharts to LikeMeDonuts, all participants preferred LikeMeDonuts. Three reasons were commonly mentioned. First, LikeMeDonuts provides the capability to track individual records (by following a radius of the circle), which is not possible in barcharts. *"The donuts show an extra level of information"* one participant explained. Second, LikeMeDonuts shows an overview of the entire peer group while barcharts only show overviews of individual criteria. One participant stated that *"when I turn off the labels and step back, I can estimate the overall similarity of the group from the colors"* and another commented that *"using barcharts, I need to read eight separate charts. I only need to focus on one chart using the donuts."* In contrast, they thought barcharts were only useful for reviewing a single criterion at a time, as one participant said: *"It makes no sense to compare the bars between two criteria, like the number of female students to the number of Computer Science students."* Finally, five participants mentioned that LikeMeDonuts is more aesthetic and one added: *"It looks cool. I feel more motivated to show this to my friend."*

The participants also pointed out that a unique advantage of barcharts is that they make visible trends in the criteria values, e.g., *"it shows me the overall shape and I can clearly see records with extreme criteria values,"* or *"barcharts can guide me to filter out outliers."* In comparison, they found it difficult to review criteria distributions in LikeMeDonuts, where criteria values are repeatedly split within each branch. *"Values in the outer rings are not aggregated and I need to review the sectors one by one,"* commented by a participant during the second round of study. To address this weakness, I coordinated LikeMeDonuts with barcharts: when users click on a subgroup in the donuts, the distributions of the selected records will be highlighted in the barcharts (Figure 4.4b). This enables users to review the criteria distributions of subsets of records.

4.3 Discussion

I discuss the limitations and new opportunities discovered in my study.

4.3.1 Limitations

All the study participants were university students, so a more diverse population should be tested to further improve the interface. Larger numbers of participants and longer periods of use may alter usage patterns and lead to new strategies and other analysis workflows. The interface can be further improved. For example, the green similarity color encoding could be applied to the timelines as well and missing data may have to be represented with a separate encoding.

Scalability becomes an issue for most interactive visualizations as the size of the data grows. My system prototype runs smoothly with a testing dataset of 10,000 records, each with an average of 40 events. A larger number of archived records can slow down the computation of similarity and the rendering of the visualizations. Better techniques to cluster and compare records in groups would enhance the performance for applications requiring extremely large datasets, such as millions of online customer records. When the number of criteria grows larger, showing all criteria at once is likely to overwhelm most users (as illustrated in Figure 4.7). Automatically selecting two or three criteria to start may be useful [89,90]. Splitting the criteria into multiple LikeMeDonuts may also be useful (e.g., one for demographics, another for academic experience, and a third for work experience), but evaluation is needed to identify and quantify benefits, and other solutions may emerge.

Applying the interface to other application domains is likely to reveal further issues. For example, I know that more advanced temporal query methods [2,3] will need to be integrated to tackle most medical applications. Other data types need to be supported, e.g., network connections between individuals [91–93]. My study mainly focused on the scenario of making critical life decisions when users demand more controls and context even at the cost of added complexity [70,71]. My designs and findings may not be applicable to recommender systems for making less critical decisions in entertainment and shopping applications.

Finally, while most students, patients, and others who must make life choices are eager to follow the paths of predecessors, there are dangers to such an ap-

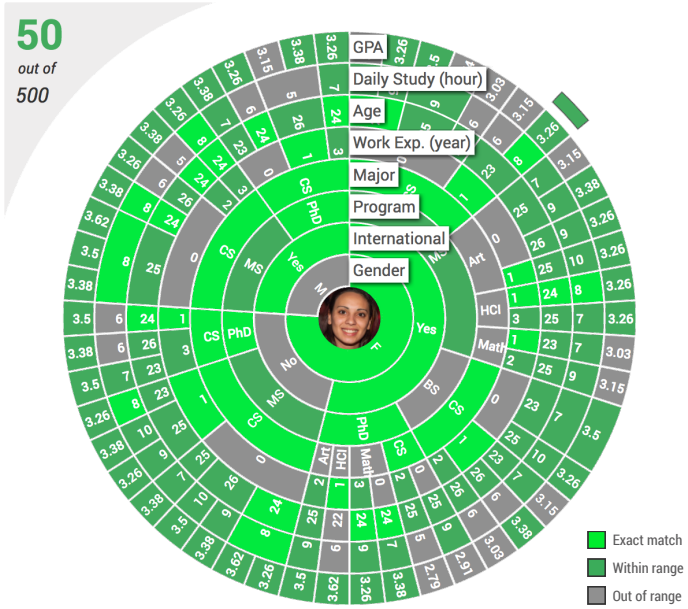

Figure 4.7: The LikeMeDonuts showing all the 8 available criteria of the student dataset used in the usability study.

proach. Biases may be introduced when the data available do not represent people adequately or when there are errors or missing attributes in the data [94]. 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. Visual analytics solutions may already be a big improvement compared to black box solutions, but how do we provide guard rails to limit the effect of possible biases?

4.3.2 New Opportunities

While automated black-box recommendation techniques are effective and used widely in entertainment and shopping applications, transparency is critical when

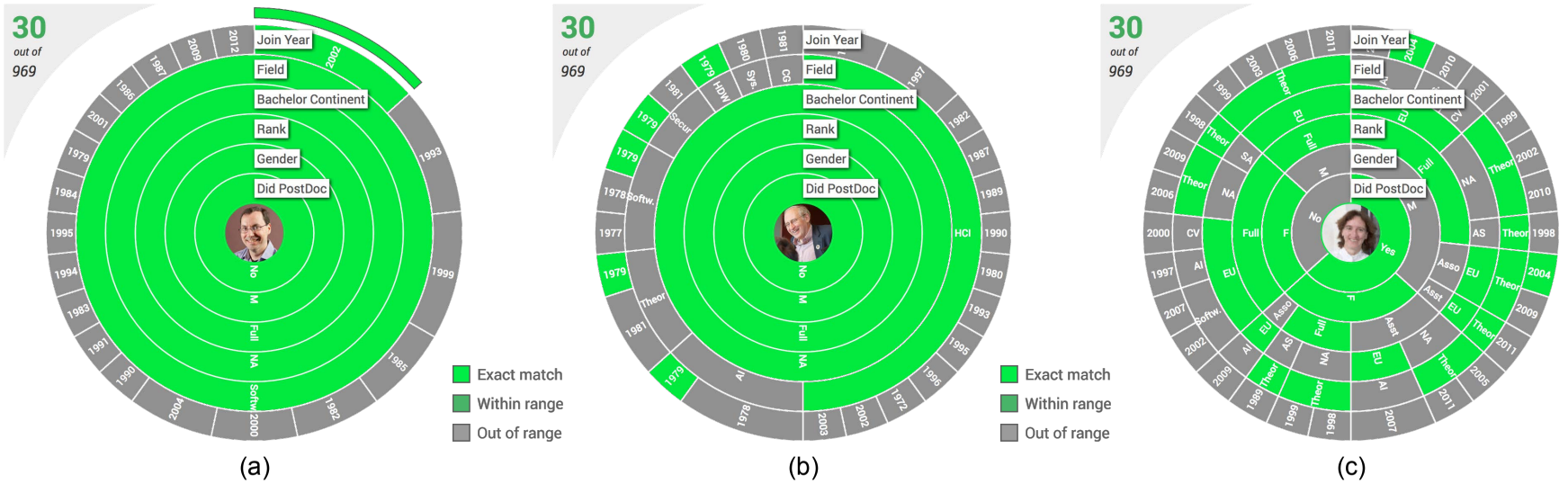

Figure 4.8: This example uses a real dataset of 969 professors from top Computer Science Graduate Programs [12]. The three LikeMeDonuts visualizations show the top 30 most similar records of (a) Dr. David M. Brooks, (b) Dr. Ben Shneiderman, and (c) Dr. Claire Mathieu. The most similar records of Dr. Brooks are all identical to each other except for joining year. Dr. Shneiderman is unique in research field and joining year compared to his peer group but normal in other criteria. The peer group of Dr. Mathieu is very diverse for all criteria.

users review recommendations for critical life decisions [31, 71]. My early investigation suggests that visual representations such as the LikeMeDonuts can help users review similarities and differences in the peer student group. Another example with a real dataset of professors is shown in Figure 4.8.

Beyond similarities and differences, "DiverseDonuts" can also be designed to guide the creation of diverse teams. Diversity can drive innovation in teams [95]. 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 diverse divisions of the company, genders, backgrounds, and with some age and background variations. Detecting clusters and selecting representative records from each cluster is a potential approach to pursue.

Finally, I believe that tools such as the one described in this dissertation can help data scientists define better distance metrics that can then be used automatically in some situations after proper evaluations are conducted.

4.4 Summary

Recommendation applications can guide users in making critical life choices by referring to the activities of similar peers. This chapter focused on how to improve the selection of peer groups. It described a novel set of visual techniques (LikeMeDonuts, Ranking Glyph, and History Heatmap) and a visual encoding of similarity, which can be combined with basic methods for criteria selection and timeline views.

The resulting combination and user-controlled selection of workflows enable users to rapidly evaluate the similarities and differences in a peer group compared to a seed record. Interaction facilitates the review of aggregated summaries as well as individual record views and their ranking. A formative lab evaluation strengthen my belief that finding "people like me" is a challenging problem that will greatly benefit from visual analytics approaches. While similarity between people will remain a subjective measure and vary based on the context of use, the creation of ground truth datasets for specific situations will pave the way to more formal evaluation.

Chapter 5: Event Sequence Recommendation: Workflow, Interface, and Integration

The growing interest in event analytics has resulted in a flurry of novel tools and applications using visual analytics techniques to tackle varied problems in healthcare, customer service, education, cybersecurity, etc. The central tasks include describing, summarizing, or comparing collections of event patterns, searching event sequences to find records of interest or build cohorts, predicting outcomes associated with event patterns, studying variants from established workflows, etc. I believe the next breakthroughs for event analytics will come by going beyond the usual descriptive and predictive analytics to develop actionable guidance by way of prescriptive analytics [96].

In layman's terms, the prescriptive analytics for event sequences consists of recommended actions (what and when) that would lead to the desired outcome based on the history of similar archived records. 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, I look at those who became professors, review the recommendations, and discuss together an action plan, combining the wisdom of the advisor and the system's recommendations based on events and timings identified as correlated with becoming a professor.

The research question is *what combination of algorithmic analysis and interactive visual exploration can augment analysts' ability to review recommended actions and improve outcomes?*

Recommender systems are being widely used to assist people in making decisions, for example, recommending films to watch or books to buy. The main novelty of the approach proposed in this dissertation is that it uses event sequences as features to identify similar records and provide appropriate recommendations. While traditional product recommendations can be described with simple explanations such as "people with attributes like yours also looked at this product or watched this movie," my approach can be summarized by the following statement: "Based on what happened to people who started with an event sequence similar to yours, what the sequences of actions and their timings are that might lead to your desired outcome."

Properly presenting and explaining recommendations is critical to the effec-

tiveness of recommender systems and decision support tools in general, as it helps develop users' trust in the system and motivate users' actions [97]. Visualization techniques, such as ranked lists [45] and two-dimensional maps [98], have been used to pursue this goal. EventAction[1] provides a visual analytics approach to (1) find similar archived records, (2) explore potential outcomes, (3) review recommended event sequences that might help achieve the users' goals and identify key steps that are of particular importance, and (4) assist users as they interactively define a personalized action plan associated with a probability of success. The main contributions of this dissertation are as follows:

- The first attempt—to the best of my knowledge—at a prescriptive analytics system to present and explain recommendations of event sequences.
- A proposed four-step workflow for event sequence recommendation.
- A design study of EventAction, which instantiates the proposed workflow in the context of a student advising application, and reports on an evaluation conducted with three graduate students.

The general EventAction principles instantiated in the student advising application can be applied to many other domains. In the case of doctors formulating medical treatment plans, EventAction can help doctors find archived patients who have medical histories similar to the current patient and identify treatments associated with a good outcome. Another application might be eCommerce companies

[1]This work was published at IEEE VAST 2016 [67].

planning a series of interventions to retain a current customer. They would find archived customers who started with an event sequence similar to the current customer, and then recommend sequences of actions and their timings that increase the likelihood of retention. A third promising domain is sports coaching. For example, in the middle of a basketball game, a good coach formulates a plan to increase the team's likelihood of winning the game. EventAction can help the coach find archived games that had a similar first half, and suggest actions such as using an agile point guard immediately or attempting more three-pointers in the last five minutes.

5.1 Preliminary Design of EventAction

5.1.1 Driving Application and Needs Analysis

The new concept of EventAction had been germinating based on prior event sequence analytics case studies. The design process was accelerated by choosing a specific application (student advising) to drive a multi-phase design study. My process was inspired by the nine-stage framework proposed by Sedlmair et al. [99]. Specifically, my work 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. This section focuses on the discover stage, while later sections cover the design, implement, deploy, and reflect stages, which informed revisions to the user and task characterizations, and led to refinements to the prototype.

To learn about student academic planning, I worked closely with the professor who manages the Computer Science department's review of graduate student progress and has eleven years of experience in student advising. I will call this main category of target user the "review manager." The department conducts annual reviews of students' accomplishments to encourage progress through program milestones. Students report their activities during the past year, including the series of courses they took, papers they published, internships, awards, etc. Based on these event sequence data, the review manager conducts one-on-one reviewing sessions with the students to provide recommendations and help them plan the subsequent years so they may reach their career goal.

Often, the review manager makes recommendations by referring to the department's requirements and by recalling the experience of students he advised in the past. While certain general recommendations such as *"finishing your classes no later than the fifth semester"* or *"starting to work with professors in the second year"* can be made in this manner, the review manager found it difficult to personalize the recommendations to fit each student's progress and career goal, and finding relevant stories from past student histories that may provide inspiration and encouragement.

Facing this challenge, the review manager needs a tool to help him analyze the collected dataset of archived students' academic activities, and augment his ability to make personalized recommendations for each student. I held weekly meetings with the review manager during which I conducted informal interviews to understand the advising workflow and demonstrated the early prototypes of EventAction to collect his feedback and suggestions. Based on the discussions, I gathered and refined a list

of design needs that EventAction should support to augment the advising workflow:

N1. *Find Similar Archived Students:* Querying the archived students' data to find those whose activities are similar to the current student in their early years in school.

N2. *Estimate Potential Outcomes:* Summarizing the outcomes of the similar archived students to estimate the outcome of the current student.

N3. *Recommend Actions:* Providing recommendations on what actions to take and when to take the actions to improve the current student's likelihood of achieving the desired outcome.

N4. *Evaluate Action Plans:* Providing immediate feedback on the action plan made by the current student and enabling the current student to review and tune the action plan iteratively based on the feedback.

N5. *Protect Privacy:* Protecting students' privacy by showing only safe aggregations and providing adequate management of access rights to the detailed information.

I identified three variant scenarios of use: (1) the review manager might use the tool independently, for example, before or after an initial meeting with a student, (2) the review manager might explore the data and review suggestions standing side by side with a student, and (3) a student might use EventAction alone or with a peer. I discuss other usage scenarios in the evaluation and discussion sections.

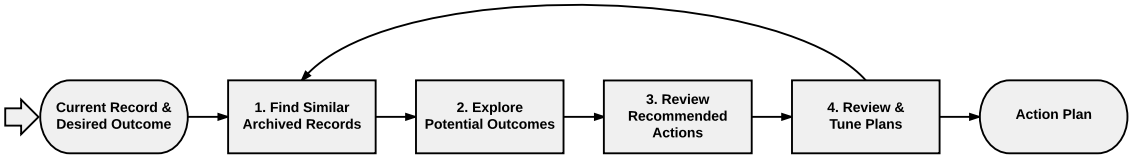

Figure 5.1: The workflow of EventAction. In this section, I provide the details of each step using the driving scenario of student advising.

5.1.2 Workflow and User Interface

EventAction enables a data-driven workflow to help analysts generate a plan of action based on recommendations[2] (Fig. 5.1). Seeded with a current record for review, EventAction extracts, from the set of all archived records, a cohort of records that are most similar to the current record. Each record is represented as a sequence of events and 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 current 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. Action plans can be made for the current record and EventAction provides immediate feedback by showing how the plan affects the outcome estimation. In this section, I describe the steps of EventAction's workflow, using the student advising scenario to illustrate those steps.

[2]A demo video is available at `http://hcil.umd.edu/eventaction`.

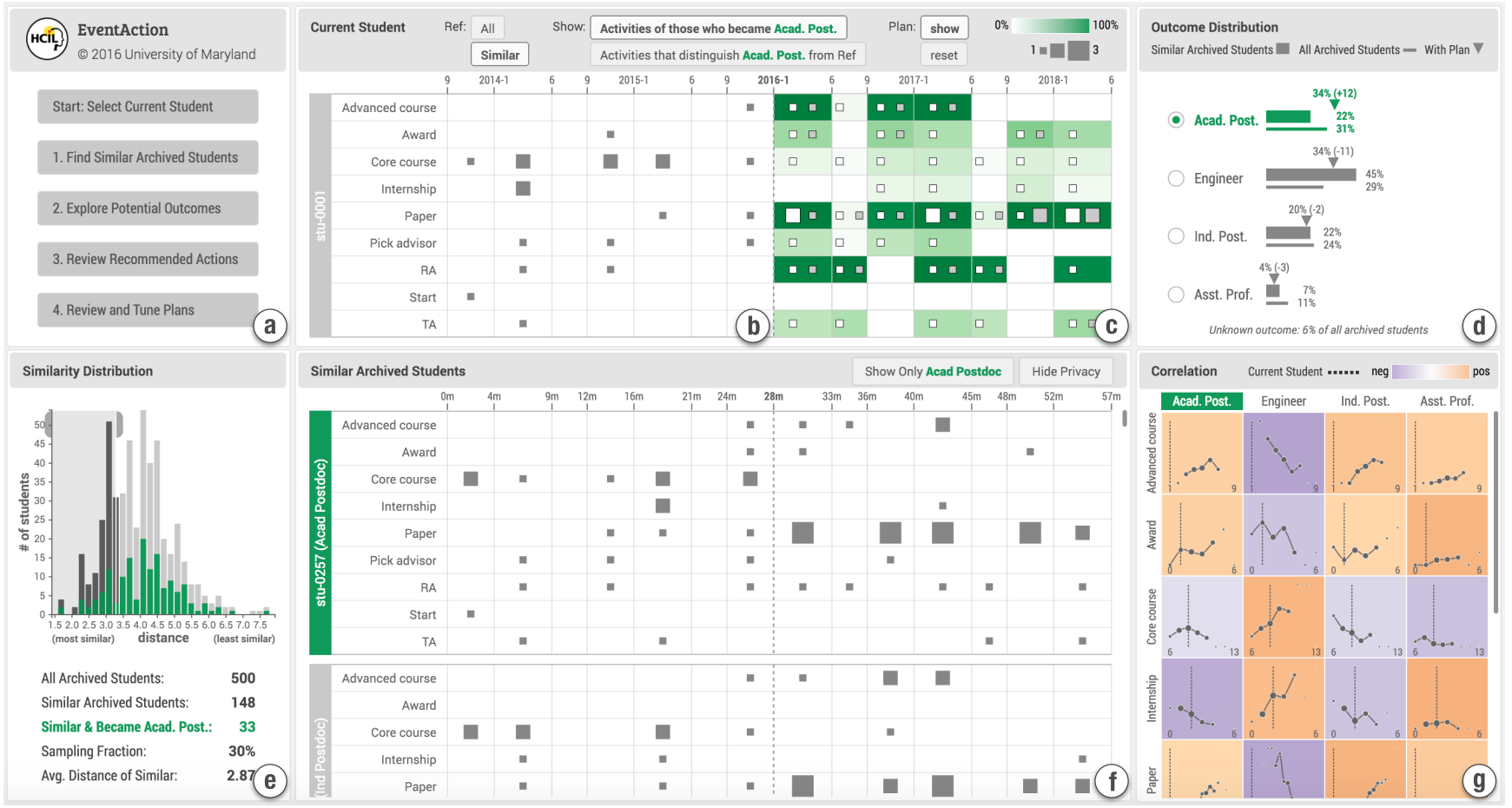

Figure 5.2: EventAction provides a visual analytics approach for helping data analysts recommend actions to improve the outcome. The user interface consists of seven coordinated views, opening progressively as the analysis progresses: (a) workflow control panel, (b) current record timeline, (c) activity summary view, (d) outcome distribution view, (e) similarity distribution view, (f) similar archived record timelines, and (g) correlation view. Figures illustrate a synthetic dataset.

5.1.2.1 Reviewing Current Record

When using EventAction, a review manager starts by retrieving a current student's record from the database. The record of a student working alone would be loaded automatically. Users can also select an initial desired outcome. EventAction shows the detail timeline in a table, where each row represents an event category and each column represents a period of time (Fig. 5.2b). To reduce visual clutter and show periodic patterns, events that occurred during the same time period are aggregated and encoded by the size of the gray square in each table cell. My initial design was derived from Lifeline2 [100]. It showed the precise timing of all events but caused overlaps when multiple events occur close together. My revised design applied the bucketing strategy [54] to aggregate the events within time periods, which dramatically simplifies the display.

EventAction allows users to specify the time periods, as they are likely to be highly dependent on specific application domains. For students' academic records, the review manager segmented each year into three periods according to the school semesters: Spring (January to May), Summer (June to August), and Fall (September to December). The time axis of the current student (Fig. 5.2b) shows the exact date, while the time axis of the archived students uses relative time (Fig. 5.2f).

5.1.2.2 Finding Similar Archived Records

To find similar archived students, EventAction compares the event sequence patterns of the current student and each archived student. The length of the compar-

ison window is defined by the length of the current student's timeline. The similarity between two students is measured by the Euclidean distance of the feature vectors extracted from the students' event sequences within the comparison window. In this dissertation, I defined the feature vector to be the number of events in each category. I chose a simple similarity algorithm to facilitate my goal of rapidly building a deployable prototype including all the steps of the workflow. The discussion section reviews possible enhancements.

Then, EventAction computes a similarity score between the current student and each archived student and shows the results in the similarity distribution view (Fig. 5.3a). I included a range selection widget to allow users to customize the set of archived students to be considered as the similar cohort. EventAction facilitates the range selection by showing five indicators which were determined through iterative refinement with the review manager: the total number archived students, the number of selected (similar archived) students, the number of selected students with the desired outcome (visible in green), the sampling fraction, and the average similarity score.

After the cohort selection, individual timelines of the similar archived students are displayed for inspection in the lower middle section of the screen, if the user has access rights to those records (Fig. 5.3b). The design partner chose to align each record by Fall, which is the typical semester for starting school. Temporal patterns such as the number of courses students take or the most common semester students advance to candidacy become easier to observe.

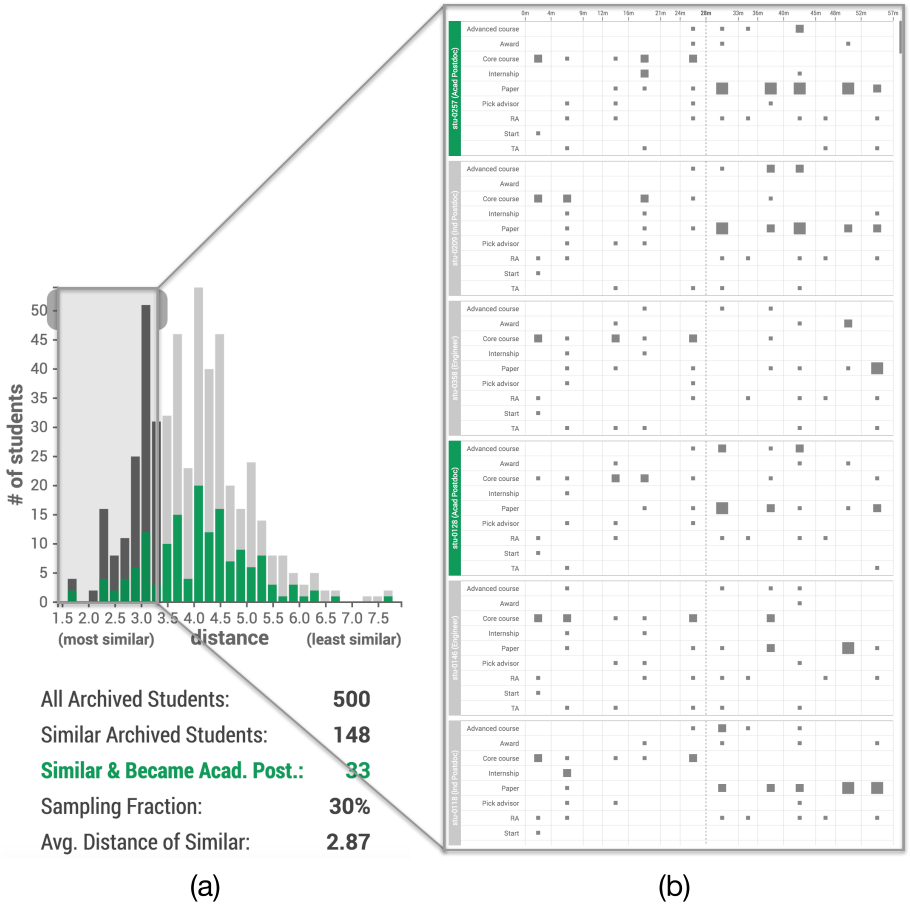

Figure 5.3: (a) The distribution of the similarity between the current student and each archived student. (b) The timelines of the selected students are displayed for inspection.

5.1.2.3 Exploring Potential Outcomes

Based on the outcome distribution of similar archived students, EventAction lists the potential outcomes for the current student and estimates likelihoods. The outcome distribution view (Fig. 5.4) shows two sets of bars: the thicker bars represent the similar archived students, and the thinner bars represent the baseline of all archived students. From this view, users can estimate: (1) the current student's most likely outcome, (2) the current student's probability of achieving the desired outcome, and (3) whether the current student is more or less likely to achieve the

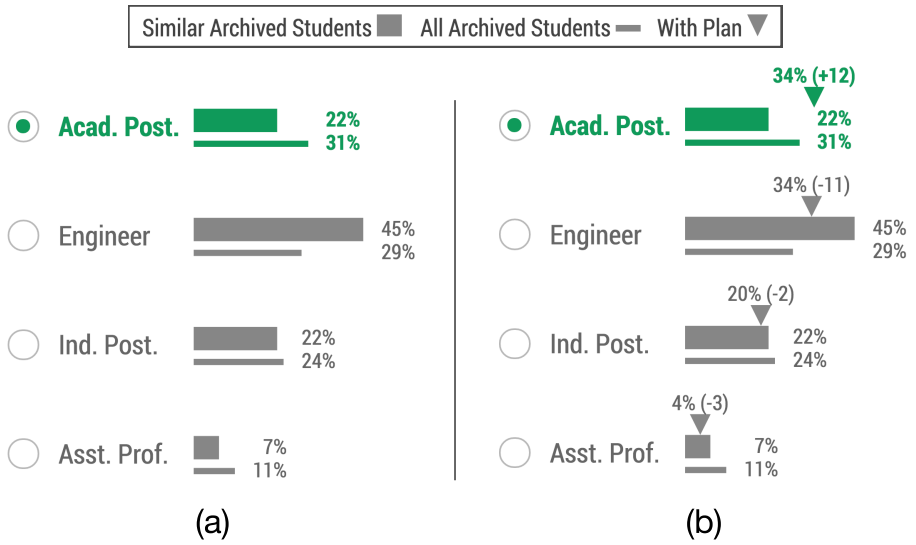

Figure 5.4: (a) The outcome distributions of similar archived students (thicker bar) and all archived students (thinner bar). (b) EventAction estimates users' action plans and show the updated outcome distribution with triangles. The desired outcome is highlighted in green.

desired outcome compared to all archived students. Users can change the desired outcome at any time in the process and all views are updated accordingly.

Using the correlation view (Fig. 5.5), users can further explore which event categories are most correlated with the probability of having each outcome, so as to identify important event categories that the current student should pay attention to when making the action plan. Each cell or line chart shows the correlation between an outcome and an event category generated based on the similar archived students. The x-axis represents the number of occurrences of that event category in a student's entire timeline. The y-axis represents the probability of having that outcome, which equals to the percentage of students who had that number of occurrences and had that outcome. The size of the dots encodes the number of records. Dots of more than 10 records are connected with lines to show the overall trends. The background

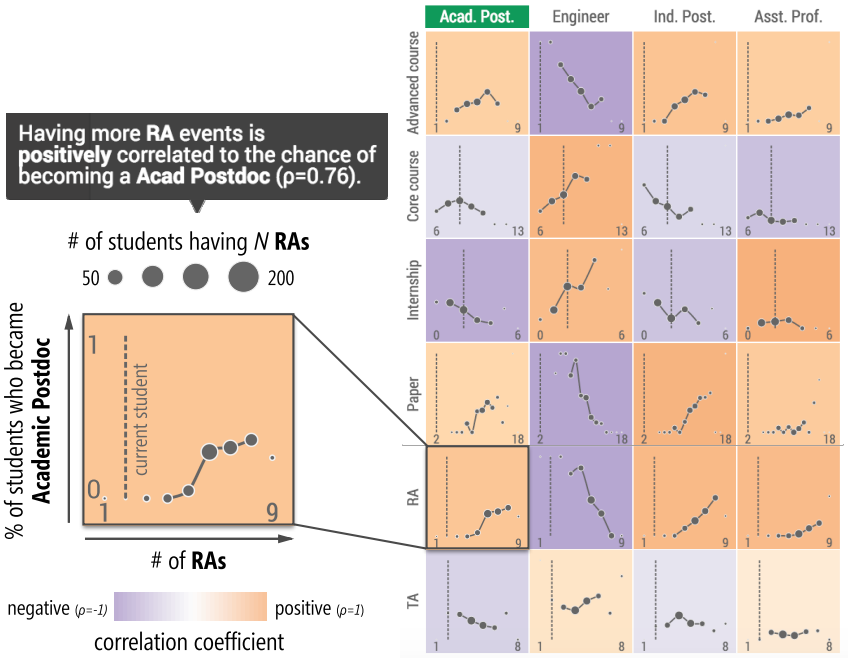

Figure 5.5: The correlations between outcomes and event categories. The enlarged example chart shows that most of the students had between 4 and 8 RAs, and having more RAs is positively correlated to the current student's likelihood of becoming an Academic Postdoc.

color of the charts encodes the Pearson correlation coefficient of the dots, weighted by their sizes. The vertical dashed line shows the number of event occurrences the current student has so far.

My initial design only used histograms to show the distributions of student populations with different numbers of event occurrences. It was named "feature distribution" but was found not very helpful. Instead of seeing only the distributions, users seemed more interested in learning how the event occurrence is correlated to the probability of achieving an outcome, especially the desired one. Thus, I calculated the percentage values for "probability of success" from the categorical outcome attribute, and added background colors to encode the correlation coefficient. I then changed the histogram to lines and dots to show the detailed relationship between

"probability of success" and numbers of event occurrences. To avoid potential misinterpretation, I added text explanations triggered by mouse hovering. I recognize that the correlation information may not be easy for every user to interpret, but its value was immediately recognized by my Computer Science design partner and students. Simpler designs may be possible.

5.1.2.4 Reviewing Recommended Actions

After identifying event categories that are most correlated to the current student's likelihood of achieving the desired outcome, users can explore the activity summary view to investigate the temporal aspect of the recommended actions. Users can choose to show either all or similar archived students (Fig. 5.6a), and can drill down to see only the activities of those who had the desired outcome of the current record (Fig. 5.6b), or compare the activities between everyone and those who had the desired outcome (Fig. 5.6c).

The activity summary view is directly integrated in the timeline of the current record (Fig. 5.6a) and the activity patterns can be used to guide the specification of the action plan. The background color of each cell in the table represents the percentage of records that had at least one occurrence of the event category in that time period. The darker the background color, the more prevalent this event category is in this time period. The size of the gray square encodes the most common number of occurrences, which suggests the typical number of this event in this time period. Users can hover on a square to review the detailed distribution of event

occurrences.

My square-based design was inspired by previous work in network comparison [11, 101], which studied different glyph designs for matrix visualizations and found that the square-based method outperformed the rest. My early prototypes also tried to color the inner square instead of the entire cell. However, this approach makes it difficult to read the color when the square is small. I also considered swapping the mapping, using the background color to represent the number of occurrences and square size to encode the prevalence, but this was inferior to my final design because the visual encoding became inconsistent with the timeline view and my users found the color less precise in representing sparse numbers.

5.1.2.5 Reviewing and Tuning Plans

After reviewing the activity summary, users can iteratively specify an action plan with the guidance of the activities of the reference. They can add events of a category and in a time period by clicking on the corresponding cell of the student timeline (Fig. 5.6d). The planned events are shown as squares side-by-side with the recommended ones and multiple clicks rotate through the range of possible values. The current design was chosen for two main reasons. First, the square-based glyph is simple and consistent with the timeline and activity summary views. My users were able to understand its meaning immediately. Second, compared to designs that encode only the difference (i.e., where the user plans less or more activities than others), the side-by-side squares give users a more direct overview about the

Figure 5.6: (a) Activities of similar archived records, (b) activities of records that were similar and achieved the desired outcome, (c) activities that distinguished records that achieved the desired outcome (i.e., the difference between (b) and (a)), and (d) users making actions plans with (b) as a reference. The background color of each cell encodes the percentage of records that had at least one event in the period, and the size of the square within the cell shows the typical number of occurrences.

current action plan. It also encourages users to personalize their plan instead of making an "average" plan.

EventAction reruns the workflow to update the outcome estimation periodically (every second by default) as the plan is being updated. Practically, EventAction adds the planned events to the current student's record, extends the comparison window accordingly to the new length of the current student's record, and updates the cohort of similar archived students. Finally, EventAction updates the outcome estimation and shows the changes in the outcome distribution view as triangles (Fig. 5.4b), giving users immediate feedback on how their action plans affect the estimated likelihood of achieving the possible outcomes. In this manner, the users can iteratively refine the action until they are satisfied with the results. I chose not to update the views of the similar archived students (lower part of the screen as in Fig. 5.2e-g) continuously to keep the context stable and focus attention on the outcome estimations.

5.1.2.6 Reflections on the Design Evolution

The overall design of EventAction went through a dozen iterations over a three-month period, during which I held weekly meetings with the review manager to deploy and demonstrate the latest version of the prototype, gather his feedback, and discuss an improvement plan. I revised the placement of the seven views of EventAction until the order matched the natural progression of the task. Adding the workflow control panel was very helpful as it suggests the next possible action

(e.g., finding similar records or specifying a plan of action) and guides users through the needed steps. Views open as the analysis progresses: only the workflow control panel is open at the start, then the timeline of the selected record can be reviewed, and the similarity distribution view appears, followed by the similar archived record timelines, and the outcome distribution view and correlation view.

Aligning the timelines of the current record and the similar archived records was important, as well as clearly highlighting the time period used for computing the similarity. Again, aggregating the data by user-specified periods (semesters in this example) both simplified the displays and facilitated the definition of the plan. One important design decision I made was to deliberately avoid suggesting a single recommended series of actions, but instead provide an environment to help users understand the basis for the recommendation and a visual representation of the actions others had taken (like trails in the sand).

Several iterations also led to the consistent use of green color for the desired outcome across different views. Only the correlation view uses a different color palette, mapping a warm orange color hue for positive correlation and cool purple color hue for negative correlation. I made this exception for two reasons. First, if I use green, then only the column that represents the desired outcome should be colored in green while others should not. Thus, I would have to use two color schemes to encode the same information in the same view, which is confusing. Second, the correlation has both negative and positive values. Thus, a bi-color scheme is necessary.

5.1.3 Evaluation

I conducted an exploratory evaluation of EventAction to understand whether and how it was helpful in student advising, and identified its usability issues and limitations. My evaluation goals were aligned with the workflow of EventAction:

- **Find Similar Archived Students:** Was the meaning of similarity clear? Were there alternative approaches to assessing similarity? What were users' strategies for selecting similar archived students?
- **Explore Potential Outcomes:** Was the outcome estimation based on similar archived students reliable to users? Was the correlation view easy to understand? How would the correlation view assist in making action plans?
- **Review Recommended Actions:** Was the activity summary view easy to understand? Would users be able to identify recommended actions?
- **Review and Tune Plans:** How did users proceed to define their action plans? How often should the outcome estimation be recalculated?

In an academic context, to protect the privacy of prior student records and ensure an accurate understanding of the limitations of the data, allowing students to work alone may be infeasible. Nevertheless, I decided to use this scenario as a usability study to guide the design of EventAction. I again use the problem of graduate student advising, but for this test scenario, I constructed a synthetic dataset of 500 archived students, and included features of the real data.

Most of the archived students were enrolled in the Ph.D. program, and their recorded event categories included "start school", "advanced course", "core course", "classes done", "masters degree", "publication", "advanced to candidacy", "TA" (Teaching Assistant), and "RA" (Research Assistant). The students' first placements were categorized into four types, including (1) software engineer, (2) industrial postdoc (e.g., research positions in labs such as Microsoft Research), (3) academic postdoc, and (4) assistant professor. The placement information was used as the students' possible outcomes.

I recruited three current Ph.D. students in my department who had never seen EventAction and elicited their feedback and suggestions. A laptop computer with a 15.4-inch display was used. I asked the participants to imagine that the selected current student was them and to use EventAction to make a plan to increase their likelihoods of achieving their desired outcomes. I provided no training and encouraged the participants to think aloud and report their difficulties and any findings of interest. Each session lasted about 50 minutes. The timeline view showing the records of similar archived students was disabled, just as it would need to be when using real data, in order to protect privacy.

All three participants (referred as *P1-3*) found the workflow control panel easy to use and followed the workflow in their analyses. Below I describe the study results from each step of the workflow.

Find Similar Archived Students: All three participants understood the similarity distribution view and discovered that they could use the selection brush to adjust the cohort of similar archived students. The participants diverged in their

strategies for selecting similar archived students. *P1* selected the first half of all archived students as similar and commented: *"The shape looks like a normal distribution so I set the threshold at the average."* *P2* selected the 100 most similar archived students: *"I only want those who are more similar to me."* *P3* explored different strategies and decided to set a threshold of a third of the largest similarity score. He explained *"setting a lower bound gives me more confidence."*

Explore Potential Outcomes: *P1* chose academic postdoc as his desired outcome, *P2* chose assistant professor, and *P3* chose software engineer. All participants verbalized that they could estimate their own likelihoods of having each outcome from the outcome distribution of similar archived students. *P1* immediately found that *"my chance seems below the average."* *P2* was concerned about the reliability of the results as he realized that *"the number of assistant professors is small."* *P3* thought the estimation could be more accurate if he could prioritize the event categories and put more weight on core courses. *"These are more relevant to my goal,"* he explained.

All participants had to spend at least five minutes to fully understand the correlation view. One common initial misinterpretation was to see the y-axis of the line chart as the number of students instead of the percentage. *P1* and *P3* corrected this misinterpretation by themselves as they inspected a few more charts, and the experimenter provided clarification after *P2* remained uncertain about the meaning of the correlation chart for five minutes. The participants found many insights after they became familiar with the charts. For example, *"I need to take more advanced courses to increase my chance"* (*P1*), *"RA and publications are important"* (*P2*),

and *"publications seem not relevant to me"* (*P3*). *P2* and *P3* expressed concerns about the large number of charts that need to be inspected. *P2* explained *"it is hard to keep track of what I have found, ..., I want a summary statement to remind me of the important things."* *P3* suggested sorting the event categories by their correlations to the desired outcome: *"I want to see the important ones first."*

Review Recommended Actions: All three participants were able to understand the activity summary view without training. They started by reviewing the activities of both all and similar archived students and found patterns and outliers, such as *"students take more advanced courses than core courses in the later years"* (*P1*), and *"some students pick advisors as late as in their fourth year"* (*P3*). They then narrowed down to those who had achieved their desired outcomes. *P2* and *P3* commented positively on the consistent use of green color for showing data relevant to the desired outcome: *"I know things that are green in the timeline are important and need to pay attention to."* While *P1* and *P2* understood the concept of "distinguishable activities," it took *P3* a while to realize it was a simple comparison. *"There are too many levels of subgroups and I was lost," P3* explained.

Review and Tune Plans: None of the participants noticed the table cells in the activity summary view became clickable at this step. The experimenter had to provide hints to help them proceed, and *P3* suggested providing guidance when users enter this step for the first time. When making action plans, *P2* and *P3* mainly referred to the activities of those who achieved their desired outcomes, and *P2* explained *"I want to at least be similar to these students."* *P1* primarily referred to the activities that distinguish those who became academic postdoc and said:

"These activities can make me stand out from the average." All participants used both reference groups and switched between them multiple times. They also referred to the correlation view. *"The correlation view tells me what to do and the activity summary view tells me when to do," P3* emphasized.

All three participants explicitly mentioned that EventAction's immediate feedback made them more motivated to improve the plan: *"I am not satisfied; I probably need to make a better plan," P1* said as he found his likelihood of becoming an academic postdoc is still below all archived students. *"The feedback enable me to make and compare alternative plans," P3* commented.

In the end, all three participants completed an action plan. *P2* was particularly satisfied with the experience and said: *"I appreciated that EventAction is evidence based. It is easier to understand than professors' suggestion. Different professors often gave me different suggestions and confused me a lot." P1* hoped to make an optimal plan and proposed a feature that *"I only need to set my expectation and EventAction tells me what to do." P3* expressed concerns about the reliability of EventAction's approach that *"the [archived] students might graduate many years ago and things have changed a lot today."*

In practice, until the accuracy and value of the recommendation and outcome estimation have been validated, it is unlikely that students would interact directly with private data about other students or that students would evaluate the likelihood of outcomes in the absence of guidance and encouragement of an advisor. However, this evaluation step still provided valuable usability information and input from students.

5.1.4 Discussion

My early evaluation suggests that EventAction was helpful for students, as they were able to use EventAction to effectively find similar archived students, explore the potential outcomes of the current student, review recommended actions, and prepare and iteratively improve action plans. Overall, the prescriptive analytics workflow of EventAction was easy to learn and the data-driven approach to student advising was appreciated by users.

5.1.4.1 Reliability of Recommendations

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. On the one hand, the reliability depends on the quantity and quality of the data available. To better profile the current advisee 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. In the early design of EventAction, the data contained only event sequences and outcomes. Additional attributes for records (e.g., demographics information) and events (e.g., the grade of a course) can be included to improve the quality of the retrieved set of similar archived records.

On the other hand, convincing users that the recommendations are actually reliable may prove to be equally difficult, and will require further research on the impact of algorithms and the user interface on users' perception of the quality of the

prediction. Overconfidence can also be an issue. My users identified several promising elements in the design of EventAction: (1) visually presenting the raw data and the statistics, (2) consistent use of color to mark data and patterns relevant to users' desired outcomes, (3) providing detailed textual explanations on demand as tooltips, and (4) presenting not only unexpected insights but also expected findings that match the users' domain knowledge. Users also pointed out several limitations of EventAction. For example, my initial similarity algorithm does not give users the flexibility to tune the similarity measures, and it is difficult to keep track of findings and recall them at the plan-making stage. Besides, although users could open multiple windows to make multiple action plans in parallel, my current prototype does not support saving or visually comparing alternative plans, which would be a useful feature to add.

Compared to the recommendations of products to purchase or films to watch, recommendations using event sequences could yield an exponential number of possible combinations, and differences between two recommended event sequences are likely to be small. The novel approach EventAction uses to solve this problem is that it does not explicitly recommend a particular sequence directly (e.g., Shani et al. [102]), but relies on the user to interpret the probabilities from the correlation analysis and aggregated event sequence on a timeline to construct a reasonably good, even if not optimal, plan.

5.1.4.2 Scalability and Generality

Scalability remains a challenge for most interactive visualizations. This initial prototype did not tackle scalability issues yet. It ran smoothly with a testing dataset of 10,000 records, each with an average of 42 events. A larger number of archived records would slow down the searching for similar archived students and the automatic re-computation after the action plan was updated. A manual mechanism could be used instead to allow users to decide when to trigger the time-consuming functions. For applications requiring extremely large datasets, such as millions of web customer records, interactive tools using EventAction's current workflow may help researchers understand the role event sequences can play in determining similarity and selecting a plan of action, and ultimately lead to specialized non-interactive algorithms for real-time action selection (i.e., determining a series of interventions).

Finally, the student review application I selected offered useful simplifications allowing the rapid development of a functional prototype that could be deployed for immediate testing. Graduate student records tend to be of similar length, the number of event categories is fairly small, and the semester organization lends itself to a meaningful bucketing strategy to simplify the display of temporal patterns. Easy access to real data and experienced users was also a significant advantage, and my pre-existing familiarity with the general domain and data contributed to my ability to design a useful interface rapidly.

While I believe other application domains will benefit from the general approach of EventAction, further research is needed to tackle the wide variety of event

data characteristics and the needs of different users. To start this process I have initiated a collaboration with eCommerce industry partners to investigate the use of EventAction to plan multi-step interventions. My early discussions suggest that a potential use for EventAction is to help in-house analysts devise and tune the strategies to find similar customers and plan interventions that match the desired outcome (e.g., retain a customer or get him to upgrade) with the goal of later transferring those strategies to automated algorithms. I have also started investigating applications in the medical domain.

I believe my approach will provide a fresh way for doctors and researchers to plan long-term medical treatments and follow-up actions associated with a desired outcome. EventAction's approach may facilitate the discussion between patients and medical professionals as they make choices and plan treatment next steps, and—once further refinements are made—may inspire new ways to provide evidence-based medicine and foster patient engagement in the decision process. My preliminary studies with health data suggest many specific needs. First, interval events (e.g., a week-long hospitalization) need to be treated differently from point events (e.g., a blood test) since the event duration is often critical to making decisions. Passive events (e.g., disease symptoms or diagnoses), which users cannot plan for, should be tagged separately from active interventions (e.g., treatments). Furthermore, users need to be able to prioritize certain events in the records and ignore others—such as those coming from untrusted sources. Finally, records are typically long and complex, so finding a similar case may rely on matching complex patterns but focusing on a small portion of the record.

5.2 Automatic Recommendation of Event Sequences

In this section, I introduce an extension to EventAction for generating recommendations of action plans automatically[3]. 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, I describe the algorithm for generating sequence recommendations and the challenges and solutions for integrating it into EventAction.

5.2.1 Sequence Recommendation Algorithm

Markov decision processes (MDPs) are widely used in applications as a mathematical framework for solving sequential decision problems (e.g., navigating a robot) [104]. My sequence recommendation algorithm developed based on MDPs and an implementation provided by Theocharous et al. [105]. This section provides a step-by-step overview of how the algorithm generates recommendations of event sequences (illustrated in Figure 5.7).

[3]This work was published at ACM CHI EA 2018 [103].

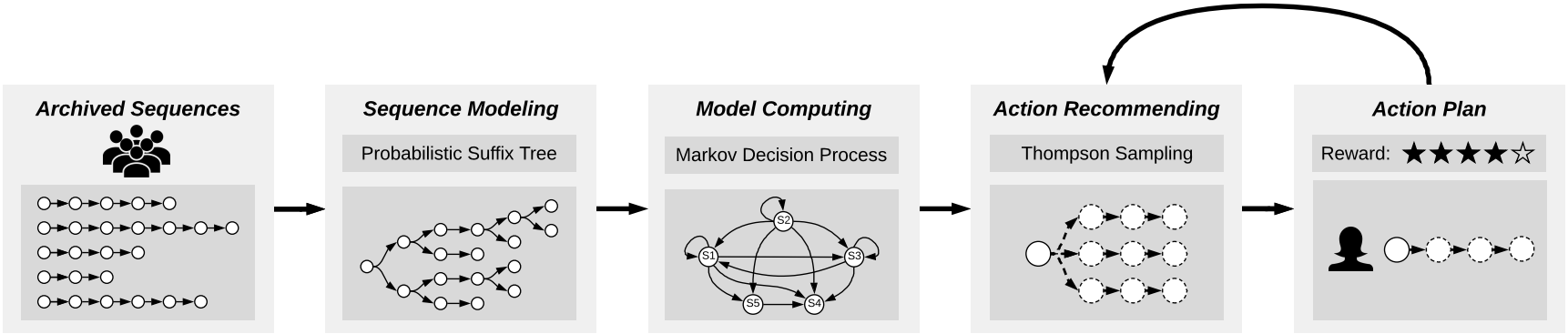

Figure 5.7: An illustration of the sequence recommendation algorithm.

5.2.1.1 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 $P(e_{t+1}|S)$. My implementation used the *pstree* algorithm [106] in R language.

5.2.1.2 Markov Decision Process

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.

5.2.1.3 Thompson Sampling

The last step is to find the optimal policies generated by the MDP models for generating recommended event sequences. My implementation uses Thompson

sampling [107], which is a heuristic approach for choosing actions that addresses the exploration-exploitation dilemma in the multi-armed bandit problem [108]. 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 [109] 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. [105] 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.

5.2.2 Integration into EventAction

I describe the three major challenges and my solutions for integrating the automatic recommendation algorithm into EventAction.

5.2.2.1 Event Co-Occurrence

The sequence recommendation algorithm [105] was originally designed for recommending travel plans, where each event represents a place to visit without any overlapping. 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.

My 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 $E = (|e_1|, |e_2| ... |e_n|)$. 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.

5.2.2.2 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[4] or Google Maps[5]). 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, I 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

[4] `https://www.tripadvisor.com`

[5] `https://maps.google.com`

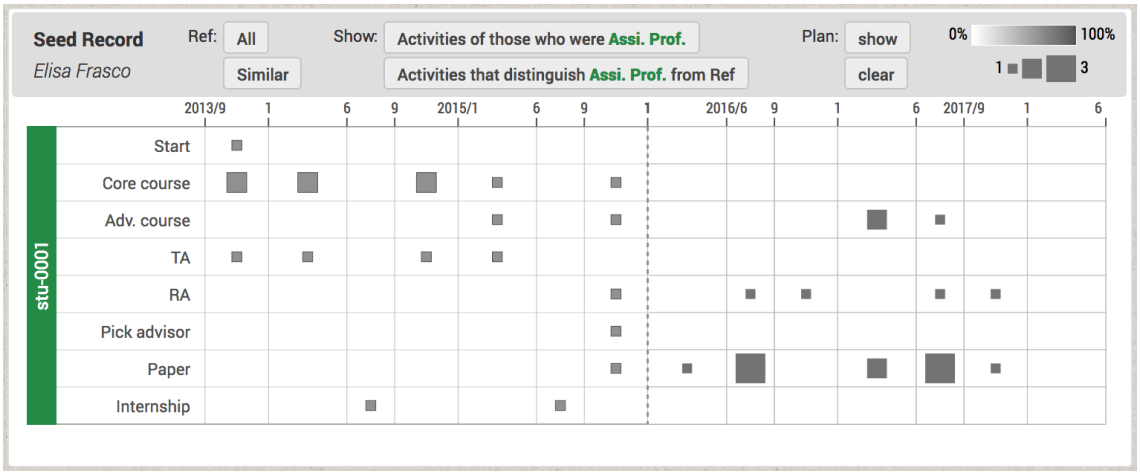

Figure 5.8: Seed record timeline (left) and recommended plan (right). In this example, the recommended plan emphasizes on research activities such as RA (research assistantship) and paper. It also suggests taking more advanced courses.

encouraged to verify this assumption or define their own reward functions.

5.2.2.3 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. 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 5.8).

5.3 Final Design of EventAction

The final design of EventAction integrates the PeerFinder visual components (Chapter 3-4) and the automatic sequence recommendation algorithm (Section 5.2), supporting a seamless analytical workflow for developing action plans to achieve the desired outcome. In this section, I describe the user interface, workflow, architecture, and data pipeline of the integrated EventAction system. I also report on an experiment evaluating its performance on large testing datasets.

5.3.1 Interface Overview

The final design of EventAction consists of two tabs of 12 interface views[6] (Figure 5.9): (a,f) seed record timeline, (b,k) activity summary view, (c,l) similar record timelines, (d) outcome distribution view, (e) outcome correlation view, (g) similarity criteria control panel, (h) similarity score distributions, (i) similar record distributions, and (j) LikeMeDonuts.

View (a)-(e) are organized in one tab for reviewing and tuning recommended plans. View (f)-(l) are in another tab for reviewing and refining similar records. All views are coordinated and connected to the same server backend. Users can switch between tabs by clicking on the navigation menu. The navigation menu also provides three configuration modes for each tab. Basic mode will hide complex views (c,e,j,k) and all user controls in (g). Simple mode will show additional views (c,k) and provide binary controls ("Use" or "Ignore") in (g). Complex mode will

[6] A demo video is available at `http://hcil.umd.edu/eventaction`.

show all available views and provide weight and tolerance controls in (g).

5.3.2 Analytical Workflow

Figure 5.10 illustrates the analytical workflow of EventAction. The workflow was developed and refined based on my 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 archived records. After submitting the similar records and the reward function, 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, adjusting the reward function to receive an updated recommendation, or refining the similar records.

I have also noticed many small deviations in the workflow during user studies and case studies. For example, some 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 analyses, 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,

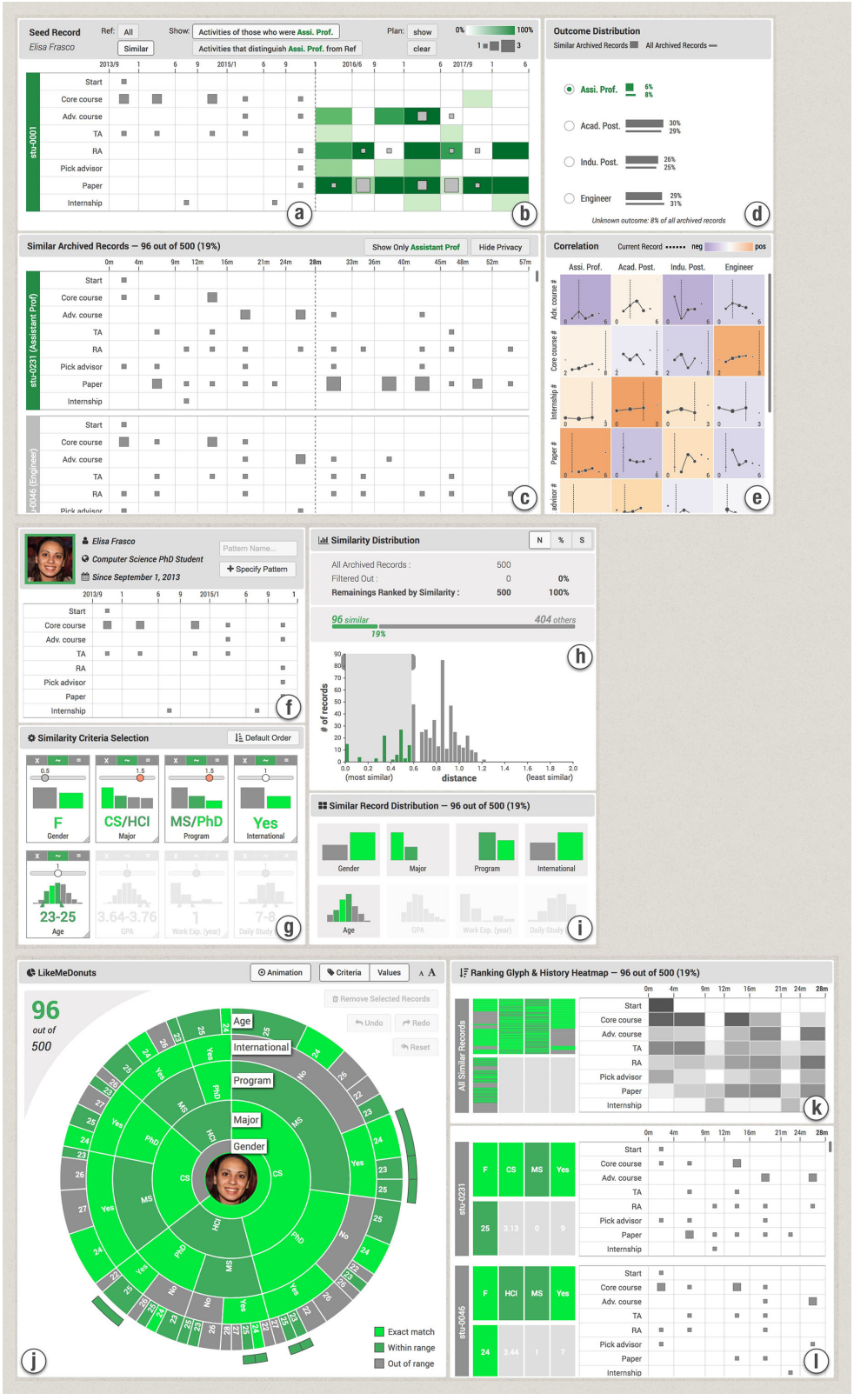

Figure 5.9: The user interface of the final design of EventAction.

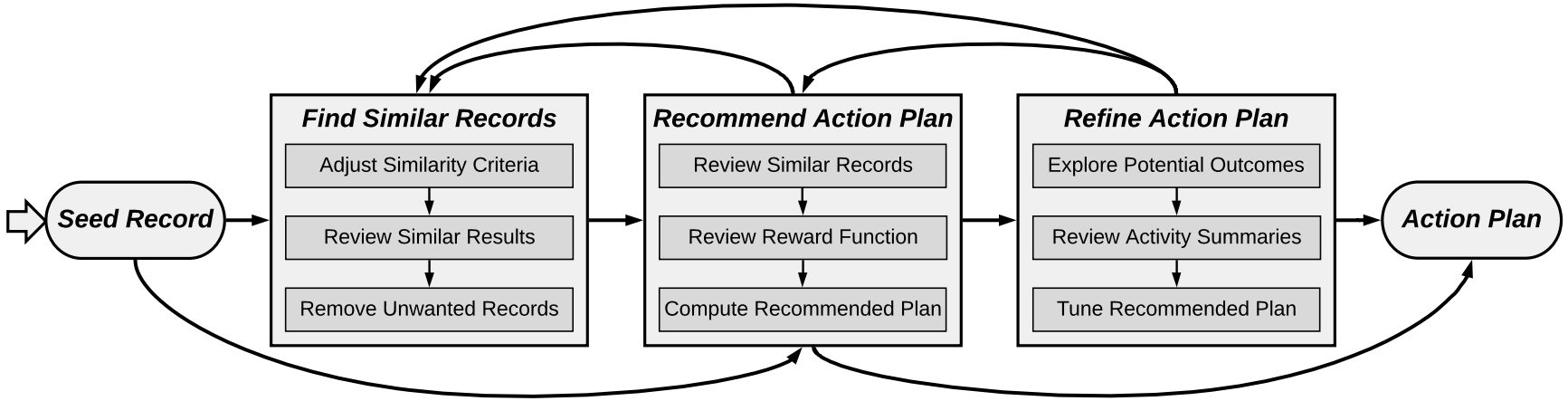

Figure 5.10: The analytical workflow of EventAction. The typical workflow starts from selecting a seed record and the first step is to find a group of similar archived records. After submitting the similar records and the reward function, a recommendation model will be computed and users can review a recommended action plan. Then, users can further refine the plan by directly editing the plan using the activities of similar records as a reference, adjusting the reward function to receive an updated recommendation, or refining the similar records.

the recommendation will be generated using a set of records retrieved with default similarity criteria.

5.3.3 System Overview

EventAction is a web application based on the client-server model[7]. The backend is developed in Python using the Flask web framework, which can be deployed remotely on a server machine or locally on a client machine. The frontend is developed in HTML, CSS, and JavaScript, and runs on a web browser. EventAction mainly used three external libraries: NumPy (scientific computing), D3.js (visualization), and jQuery (cross-platform JavaScript). This section provides an overview of the code organization, system architecture, and data pipeline of EventAction.

5.3.3.1 Code Organization

EventAction was developed using the object-oriented design methodology, where the code are organized as independent and reusable classes. The entire project consists of 13,947 lines of code, including 10,875 lines for the frontend (JavaScript), 2,209 lines for the backend (Python), and 863 lines for the automatic recommendation algorithm (R). Figure 5.11 shows an overview of the organization of the code and the system architecture.

1. **Frontend:** The frontend module contains code for visualizations, interface views, workflow controllers, and utilities. Each type of visualization is im-

[7]EventAction is available for commercial and non-commercial licensing. To request a review copy of EventAction, contact `plaisant@cs.umd.edu`.

plemented in a separate file, such as record timelines (record_timeline.js), LikeMeDonuts (sunburst_tree.js), and criteria controls (criteria_icon.js). The visualizations are encapsulated in classes and can be easily reused in other visualization applications. The interface views are built by laying out and connecting one or more visualizations into meaningful displays. The code in this group can be reused to reproduce the data displays. The workflow controllers contain application-level code for fetching data from the backend and linking different interface views together, so as to support the analytical workflow of EventAction. Finally, utilities provide helper functions that are used throughout the system.

2. **Backend:** The backend code are organized into four groups. The web server communicates with the frontend through HTTP GET and POST methods and routes the data requests. The data processing group handles loading and transforming the raw data into record instances, and storing them in memory. They also response to requests for aggregated data, such as the activity summaries for the History Map visualization. The similarity computation group aims at assessing the similarity between each archived record and the seed record based on users' criteria settings. Finally, the utility group contains frequently used helper functions.

3. **Machine Learning:** The machine learning module contains code for generating recommendations of action plans. This module runs in a separate process and communicates with the backend through inter-process pipes. This ar-

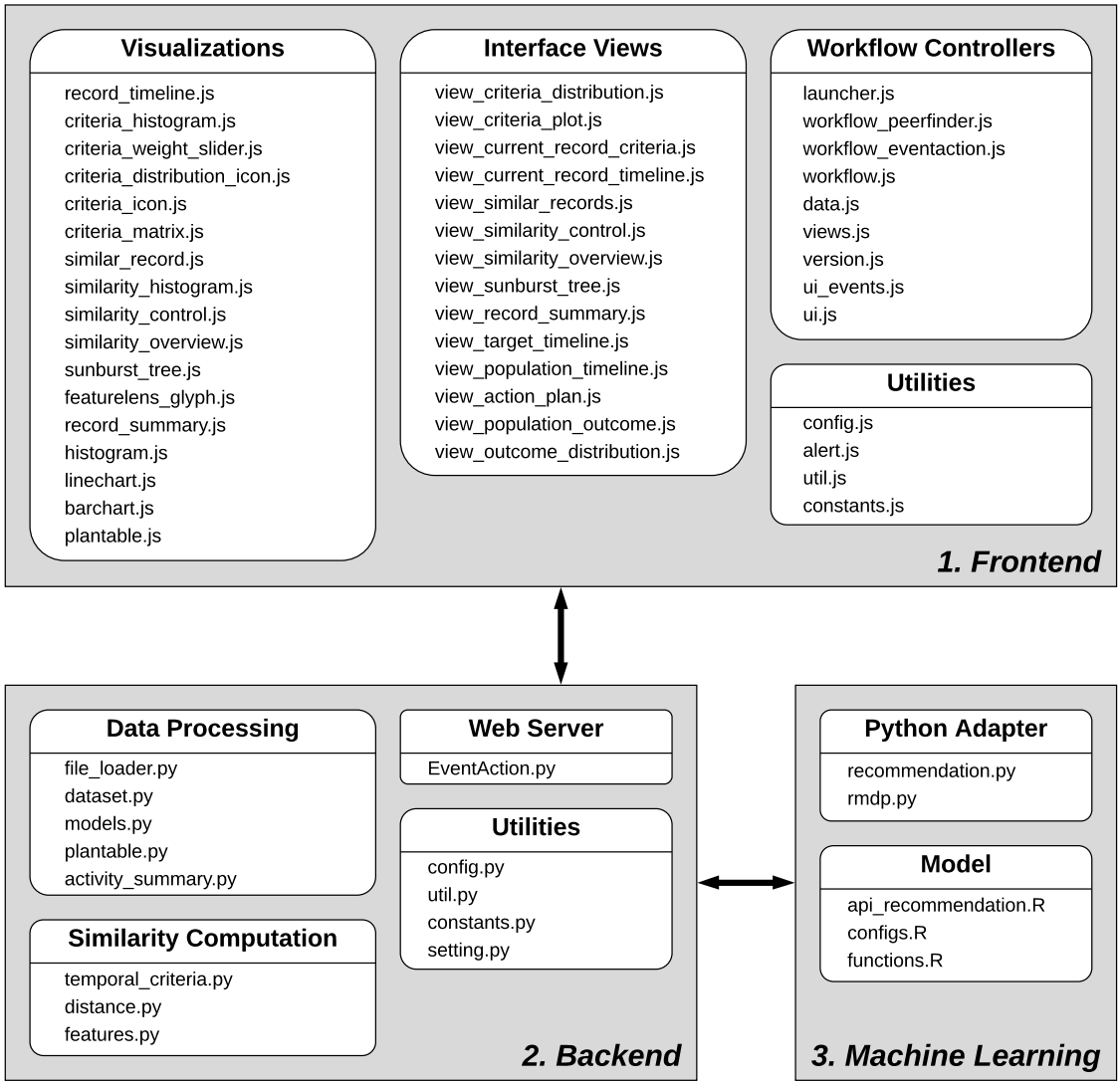

Figure 5.11: Code organization and architecture overview of EventAction.

chitecture allows the system to take advantage of a wide range of machine learning algorithms implemented in programming languages beyond Python (e.g., MATLAB, R, and Java). Moreover, separating the machine learning process from the system's main process can keep the user interface responsive during the computation, not interrupting users' exploration.

5.3.3.2 Data Pipeline

This section describes the data pipeline of EventAction for finding similar records and reports on an experiment to analyze its performance. As illustrated in Figure 5.12, 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 (Table 5.3.3.2). Each record (identified by a unique *Record ID*) is represented as a sequence of events and 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*. Next, I describe in detail how EventAction processes the data through each step of the pipeline.

Column	**1st**	**2nd**	**3rd**
Event File	Record ID	Event Category	Timestamp
Attribute File	Record ID	Attribute Name	Attribute Value

Table 5.1: Format of input files for EventAction.

1. **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.

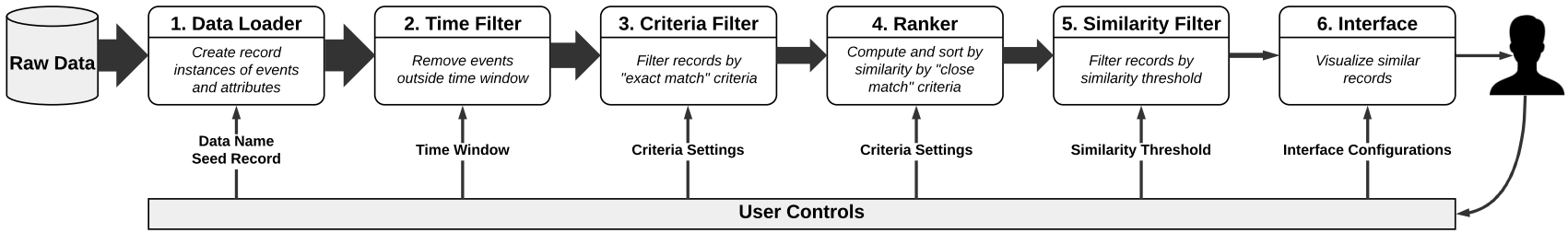

Figure 5.12: The data pipeline of EventAction.

2. **Time Filter:** At this step, users define a time window of the history, for example, from start school until the end of the third 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.

3. **Criteria Filter:** For each similarity criterion marked as "Exact Match" the following process is used: if the tolerance range is not set, only the archived records that have the exact same value (or pattern for temporal criteria) as the seed record will be retained. Otherwise, the records' criteria values need to be within the tolerance ranges to be retained. The tolerance range is represented by a set of values for categorical criteria and by a pair of upper and lower bounds for numerical or 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.

4. **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 (see Section 3.3.2 for algorithmic details). 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 algo-

rithm.

5. **Similarity Filter:** Given a Similarity Threshold specified by users, EventAction further removes records that are not similar enough compared 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.

6. **Interface:** Finally, the remaining similar records are passed to the visualization views (Figure 5.9) and shown to users. The views also provide interactive controls for users to refine the results. Users can also configure the visibility of the visualizations and controls as described in Section 5.3.1.

5.3.3.3 Performance Analysis

Finding similar records is a task that frequently repeats during analyses. I 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, I conducted an experiment using the final version of EventAction with 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,

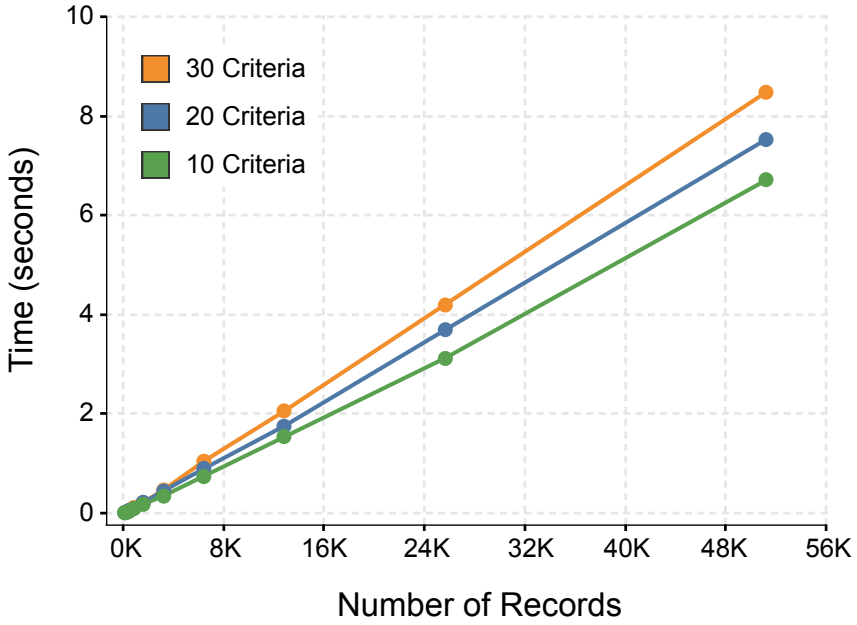

Figure 5.13: The average runtime of the EventAction data pipeline on synthetic datasets of varying numbers of records and numbers of criteria.

128,000, 256,000, 512,000, 1,024,000, and 2,048,000.

Figure 5.13 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 2, and the growth rate was mainly determined by C.

5.4 Summary

This chapter described a novel approach for prescriptive analytics that enables analysts to conduct similarity-based data-driven action planning and an automatic sequence recommendation algorithm to reduce users' effort. I designed and implemented a functional prototype called EventAction for a selected application domain

(student advising), which was tested with synthetic data for three graduate students. My evaluation demonstrated that the interface could be learned quickly and the proposed workflow was comprehensible. While recommender systems are commonly used, the novelty of my approach is that it uses both record attributes and event sequences as features to identify similar records and appropriate actions. Visual analytics techniques are particularly useful because they provide a rich aggregated presentation of the recommendations, allowing users to explore alternatives and adjust parameters. Analysts can combine prior knowledge and data-driven insights into an actionable plan along with a measure of the likely outcome. I believe that this approach can be applied to a wide variety of domains such as healthcare or business analytics, and that the dissertation opens the door to a new direction of promising research. The following chapter reports on four case studies that demonstrate the use of EventAction in three application domains.

Chapter 6: Solving Real Problems: Case Studies

This chapter reports on four case studies, summarized in Table 6, that demonstrate the use of EventAction in three application domains: education, marketing, and healthcare. Each case study was conducted with real users and using real-world datasets, following the Multi-dimensional In-depth Long-term Case studies (MILCs) procedure [110]. The case studies provided evidence of the effectiveness of generating event sequence recommendations based on personal histories, and produced five major 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. The design guidelines and usage guidelines are described in Section 7.1.1 and 7.1.2, respectively.

6.1 Students' Academic Planning for Student Advisors

This section reports on a case study conducted with a student review manager who has access to all student records. This person was a professor with 12 years of experience in advising graduate students in Computer Science. The case study took place over three weeks. During the first week, I demonstrated my prototype using a synthetic dataset and the review manager prepared a dataset of real students'

Domain	Data Size	Duration	Highlighted Results
6.1 Education	8,253 events 641 records 8 event types	3 weeks	Inspected all archived records to check the data quality. Found similar records and made an action plan for a fourth-year Ph.D. student, who wanted to become an assistant professor. Helped several students determine appropriate career goals.
6.2.1 Marketing	8,191 events 500 records 15 event types	4 weeks	Made plans for sending onboarding emails to new customers to increase engagement. Selected a seed record who received and opened the first two emails but had not clicked any links. Specified a plan with an 11% increase in the seed record's likelihood of making 1-2 clicks.
6.2.2 Marketing	26,472 events 997 records 6 event types	4 weeks	Investigated which campaign channels will be the most effective for converting a current customer into sales qualified. Selected a seed record who actively opened emails but never visited any product websites during the past 5 months. Specified a plan and the seed record's likelihood of becoming sales qualified increased by 10%.
6.3 Healthcare	3,630 events 107 records 5 event types	8 weeks	Investigated how EventAction can help health coaches prescribe personalized health interventions. Selected a current patient as the seed record, who had health alerts every day during the last three days. Specified an intervention plan and the estimated likelihood of resolving the alerts within 5 days increased by 8%.

Table 6.1: Summary of all four case studies that demonstrate the use of EventAction in three application domains.

data. During the second week, I deployed my prototype to the review manager's workstation and he used my prototype to perform the task of finding similar records. During the third week, the review manager focused on the task of making action plans. I recorded the review manager's analysis process, findings, and feedback. I provided training and necessary guidance, and answered questions over the initial visits and follow-up meetings. Figure 6.1 illustrates a synthetic dataset of student records.

6.1.1 Data Preparation

The review manager prepared a dataset of 641 archived records of graduate students in the Computer Science department. Most of the students were enrolled in the Ph.D. program. The dataset contains 8,253 events of students' academic activities, including courses (core or advanced), assistantships (teaching or research), publications, and milestones (start school, done classes, and advance to candidacy). Students' record attributes include numbers of grades (As, Bs, and Cs), numbers of assistantships (teaching and research), number of publications, class status (coursework completed or not), and candidacy status (advanced or not).

The review manager categorized the students' first placements into four types, including (1) software engineer, (2) industrial postdoc (e.g., research positions in labs such as Microsoft Research), (3) academic postdoc, and (4) assistant professor. The placement information was used as the students' possible outcomes. The review manager also had access to the records of current students.

Figure 6.1: This figure illustrates a synthetic dataset of a seed record and 500 archived student records. The seed record is a female Ph.D. student in Computer Science. The user chooses to only keep Computer Science or Human-Computer Interaction (HCI) students in either M.S. or Ph.D. program. Tolerance ranges are specified for age and Grade Point Average (GPA). More weight is given to international students. In the timeline (a) a temporal pattern for research activities was specified and added to the criteria control panel. The top 16% most similar records are selected as the peer group (c). An action plan is specified (f) and the likelihood of becoming an academic postdoc increases by 5% (g).

During the analysis the data from the author of this dissertation (a fourth-year Ph.D. student) was used as the seed record. All other records were de-identified. The analysis goal was to find a group of students similar to him, so that follow-up analyses may be conducted based on the similar records, such as predicting the first placement of the seed record after graduation and generating recommendations to help the seed record make academic plans for the next year.

The review manager worked on his own computer with a 30-inch display. He was already familiar with the interface so no training was necessary.

6.1.2 Finding Similar Records

6.1.2.1 Reviewing All Data

The review manager started with the "Show All" workflow to obtain a complete overview of the entire data. After about 5 seconds, the data were loaded and visualizations rendered. The review manager first explored the barcharts to inspect the criteria distributions of all archived students. He verified that the criteria values matched his expectations, for example, the percentages of students who had done classes and who had advanced to candidacy, and the distribution of the course grades.

Then, the review manager explored the record timelines and History Heatmap to review temporal information. He first looked at the seed record's history activities during the last four years and found 8 consecutive research assistantships since year one. *"Your research assistantship started early,"* he commented, *"this could be a*

useful pattern." The review manager also noticed two B grades during the seed record's second year of study. He specified these two temporal patterns as similarity criteria using the seed record timeline panel.

He then reviewed the temporal activities of all archived students. The History Heatmap showed an activity summary and confirmed several of his expectations, e.g., that there is a transition from teaching assistantship to research assistantship starting in the third semester, and that most students achieved the "done with classes" milestone between the third and the sixth semester as required by the department. However, two findings were unexpected. First, the students started to receive fewer As in the third semester. The review manager thought this could be caused by the increase in the difficulty of the advanced courses, or due to the fact that a number of students had finished taking classes and thus no grades were recorded. Second, nearly twice as many publication events occurred in the Spring semester than in the Fall semester. The review manager was unsure about this phenomenon. One hypothesis may be that many conferences in Computer Science announce paper acceptances in the Spring and hold the conference later in the year.

The review manager then explored the similarity score distribution. He found that the shape of the distribution had two peaks, where the first peak contained the top 37% most similar records, and the second peak was taller and contained the remaining records. *"This looks strange,"* he said, *"I was expecting a normal distribution with one peak."* To understand how the peaks were formed, he selected records in the second peak. By looking at the barcharts, he realized that those are all new graduate students in their first or second year: they all had less than four

assistantships, had not yet finished classes or advanced to candidacy. The review managed said: *"Now it makes sense. The first peak are senior students like you and the second peak are junior students unlike you. We only need those senior ones."* He then selected the top 10% most similar students and started using other visualizations to review in detail.

6.1.2.2 Reviewing Similar Records

The review manager started reviewing similar records using LikeMeDonuts. Immediately, he found that only a few exact matches were bright green while most of the sectors in the donuts were gray. *"The colors help me estimate the overall quality of the peer group,"* he commented, *"I will add some tolerance and try to make it about 50% green before reviewing in detail."* As he was adjusting the tolerance ranges, he noticed a unique branch of records in LikeMeDonuts: these records were included as the top 10% most similar but they had not done classes yet. The review manager followed the branch to inspect other criteria values of them. *"This is weird,"* he said after exploring for a while, *"they all have advanced to candidacy but not done classes. We may have errors in the data."* He clicked on the branch of LikeMeDonuts and records were highlighted in the barcharts, Ranking Glyph, and record ranked list. He reviewed the record ranked list to check the temporal activities. The review manager found that those students had not even the "start school" milestone events. He suddenly realized that they were probably transfer students brought in by professors who moved to the university. *"Their candidacy*

status was transferred to our department but some of their courses were unqualified to transfer," he explained, *"we may leave them in the results but keep the gray color to be noticeable."*

The review manager then explored the Ranking Glyph. He read the glyphs one by one and found three types of patterns: (1) green on the top and gray on the bottom (e.g., research assistantship and publications), (2) dominated by green or gray (e.g., done classes, advanced to candidacy), and (3) alternating between green and gray (e.g., course grades). *"Some criteria seem more correlated to the overall similarity and have a larger impact on the ranking,"* he commented and adjusted the criteria controls to increase the weights of research assistantship and publications, and reduced the weights of course grades. *"The alternating pattern indicates that individual course grades are not good features to characterize graduate students,"* he added.

6.1.2.3 Feedback

Overall, the review manager found the prototype very effective for finding similar students and enable a data-driven way for student advising. When asked about his preferences for the visualizations and analytic workflows, he stated that *"the three visualizations all have their own uses that cannot be easily replaced by each other."* He expressed some enthusiasm for the Ranking Glyph because *"it provides an effective overview to understand the effect of each criterion on the overall ranking."* He also liked the use of color in LikeMeDonuts because it *"provides a good*

overview of the quality of the similar records."

The review manager stated that he preferred to use the "Show All" workflow, especially when working with a new dataset: *"Starting with all the available data helps obtain an unbiased overview and provides means to check the data quality and discover initial findings to guide the analysis."* He also emphasized that *"understanding why those least similar students are different from you can also provide insights."* However, he expressed concerns about the visual clutters and interaction latency when showing all the data as the number of records becomes extremely large. In the end, the review manager applauded that *"seeing both attributes and temporal activities is important for reviewing student records. I appreciate that your system provides visualizations for this purpose."*

6.1.3 Making Action Plan

6.1.3.1 Exploring All Archived Students

In the first session, the review manager focused on exploring all archived student records to examine the quality of the data and check if the students' performance matched the department's expectation. He chose a random current student and selected all archived students in the similarity distribution view.

At first, he looked at the outcome distribution and correlation views showing the placement information of all archived students, and the activity summary view showing the activity patterns during their studies. He confirmed that the distribution of the students' placements matched his expectation and most of the activities

(e.g., courses and assistantships) met the department's requirements.

The hotspots in two event categories attracted his attention: A few students had their "start school" events in the third year instead of at the beginning. The review manager checked the source data and confirmed the pattern, explained by some students being allowed to take classes before being officially admitted. A second finding was that the most common time for advancing to candidacy was the fourth year instead of the fifth (the department's deadline) or the sixth (the effective deadline from the university, after an extension), and he commented that this provided an important insight for improving the department's management, suggesting benefits outside of the one-on-one review scenario.

6.1.3.2 Becoming an Assistant Professor

In the second session, the aforementioned fourth-year Ph.D. student in the department served as the advisee. He described his goal as wanted to become an assistant professor after graduation. The review manager used EventAction to select the top 100 most similar archived students for the analysis.

The outcome distribution showed that the most common outcome of the similar archived students was software engineer and the least common one was assistant professor. Still, the percentage of assistant professors among the similar archived students was higher than that among all archived students. The review manager could easily explain to the advisee the probability of becoming an assistant professor is low but his likelihood was above the average.

Next, the review manager explored the correlation view and looked for event categories that were most positively correlated with the assistant professor outcome, including "publication", "RA", and "advanced course". He noticed that the advisee had already been RA for several semesters but was short of advanced courses and publications. He recommended that the advisee should keep working as an RA, take more advanced courses, and start to accumulate publications.

The review manager then inspected the activity summary view to investigate when might be the best time for these recommended activities. He adjusted the controls to show the aggregated view of the activities of similar archived students who became assistant professors. The results showed a clear pattern of having an RA and publications in each Fall or Spring semester, and that the most common time for taking advanced courses was in the fourth year, before advancing to candidacy. The review manager showed the display to the advisee and they entered a draft action plan together following the pattern. EventAction estimated a 3% increase in the advisee's likelihood of becoming an assistant professor.

The review manager then switched to show activities that distinguished those who became assistant professors from others. Compared to all similar archived students, more of those who became assistant professors had TAs in the final year. The review manager endorsed the benefit of building up teaching experience before going on the job market. They refined the action plan accordingly and the estimated likelihood increased by another 2%.

6.1.3.3 Determining an Appropriate Goal

In the third session, the review manager investigated a common situation in which a current student needs help with both determining a goal and making an action plan. He picked a random current student and selected the top 100 most similar archived students. The outcome distribution showed that the current student's likelihood is above the average in becoming a software engineer, but much below the average in becoming an assistant professor. The review manager commented: *"If this student's goal is to become an assistant professor, I would recommend pursuing a postdoc first."*

The review manager repeated this process and suddenly found an outlier: the student was not similar to most of the archived students as shown in the similarity distribution view. The review manager inspected the student's record in detail and realized that the student made slow progress in both course and research: *"I need to make sure this student knows the department's requirements and deadlines."* The review manager remarked: *"EventAction could help students get a sense of their situations and help them decide whether to continue their Ph.D. studies or not."* Future development may also help identify outliers and provide support for reviewing the records before meeting with those students.

6.1.3.4 Feedback

At the end of the analysis, the review manager commented: *"Recalling a few memorable prior students and applying [the knowledge] to advise current students is*

biased. I tend to trust the data and statistics." Still, the dialog with the student suggests that the review manager was using his own judgment and experience to evaluate the value of the generated patterns and guide the recommendation process.

After the case study, the review manager planned to continue evaluating EventAction in more student advising cases. He expressed the needs to collect additional data to make the outcome prediction more specific, such as students' satisfaction toward their first placements and how soon they get promoted.

6.2 Campaign Planning for Marketing Analysts

This section reports on two case studies conducted with 5 marketing analysts and using real-world event sequence datasets[1]. 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, I provided training and necessary guidance, and answered questions. The study goal was to investigate how EventAction can help marketers prescribe personalized marketing interventions. Figure 6.2 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.

[1]This work was published at ACM CHI EA 2018 [111].

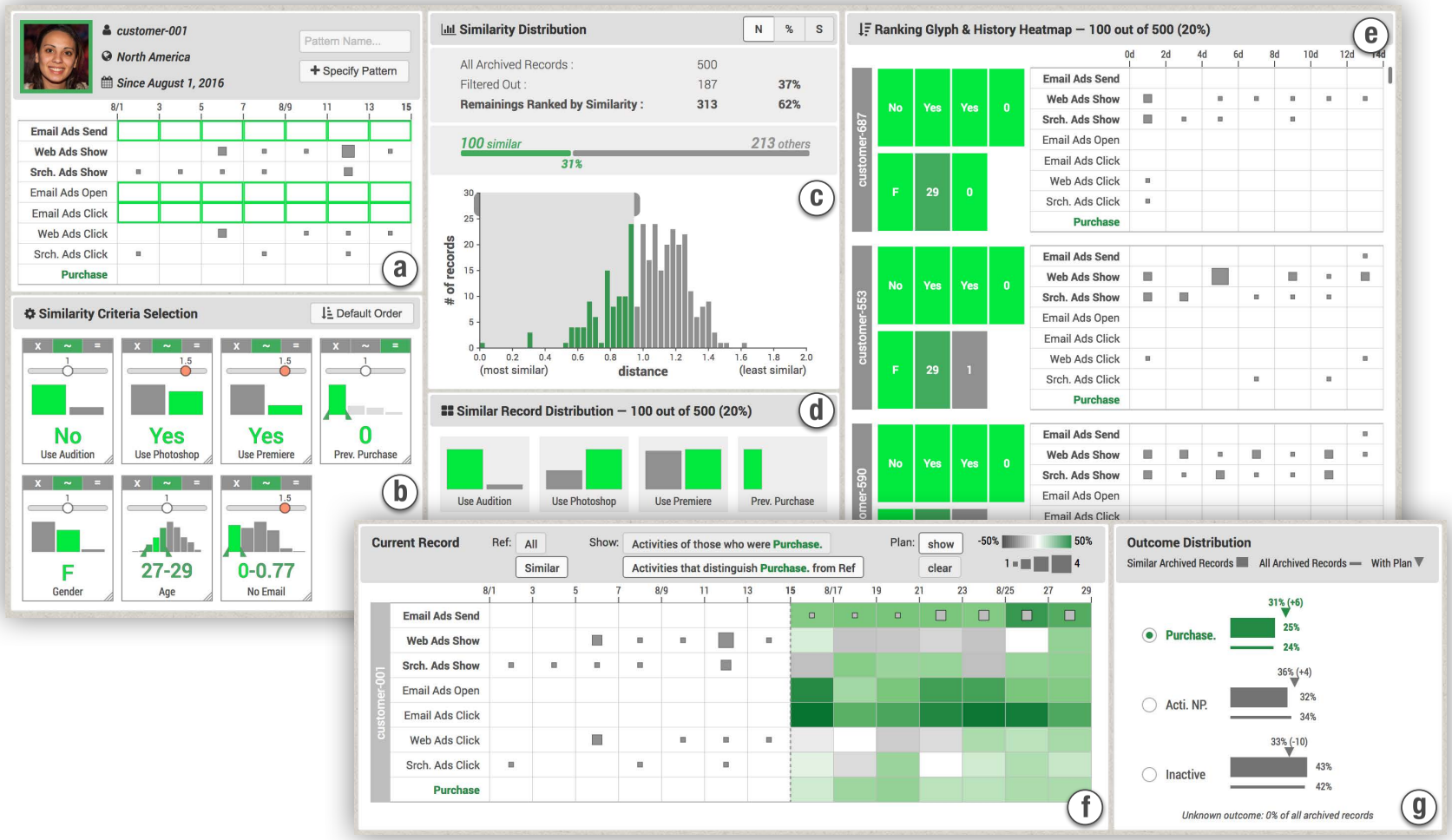

Figure 6.2: This figure illustrates a synthetic dataset of a seed record and 500 archived customer records. Marketing activities are related to sending email ads, web ads, and search ads (a). Record attributes include the customers' genders, ages, and previous product purchases (b). Three types of outcomes are defined: "Purchase", "Active but No Purchase", and "Inactive" (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 100 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 purchase increases by 6% (g).

6.2.1 Customer Onboarding

After customers start the trial of a product of the company, 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 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.

6.2.1.1 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. I 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.

6.2.1.2 Analysis

The analysts selected a seed record 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.

6.2.2 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.

6.2.2.1 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 sent". Customers' activities included "email open", "email click", and "website visit". The outcome was defined by whether or not a customer became sales qualified judged by the sales team.

6.2.2.2 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 sent", and "email 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.

6.2.3 Feedback

6.2.3.1 Pseudo A/B Testing

In both case study, the marketing analysts found EventAction useful for testing hypotheses based on historical data. They commented that EventAction allowed them to simulate plans and get results immediately, which can help select variables for A/B testings.

6.2.3.2 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 because it not only informed them about which channels were important but also showed how the importance evolves over time. In addition, EventAction enabled them to filter the records using

temporal patterns, which helps getting more precise results.

6.2.3.3 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 5.3.2).

6.2.4 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 uniquenesses of customer records and specific marketing tasks. I cover the 4 major challenges and discuss my solutions.

6.2.4.1 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.

6.2.4.2 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 amounts of events in categories are very different, ranging from hundreds of email sents to only one or two purchases; (3) many events occur at the roughly same time causing severe overlaps and visual clutters.

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. I addressed this issue by using a power scale $size = sqrt(num)$ when the range of the sizes is large. I also grouped the event categories into three classes 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 A1.

6.2.4.3 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.

6.2.4.4 Slow and Expensive A/B Testing

Conducting A/B testings 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 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).

6.3 Medical Intervention Planning for Health Coaches

This section reports on a case study conducted with two health analysts and using real-world patient health records. The entire case study lasted about two months consisting of biweekly discussions, interviews, data preparation, system deployment, and data exploration. I provided training and necessary guidance, and answered questions over the meetings and interviews. The study goal was to investigate how EventAction can help health coaches provide personalized health inter-

ventions. Figure 6.3 illustrates a synthetic dataset of patient records.

6.3.1 Task

This case study was conducted with health analysts partnered with a patient management company. The company hires health coaches to monitor patients' 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 can 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) serving as a continuity figure. While health coaches have always had to interpret information and decide 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 one to make these decisions based on what worked for similar cases, offering new found possibilities for precision healthcare through mHealth.

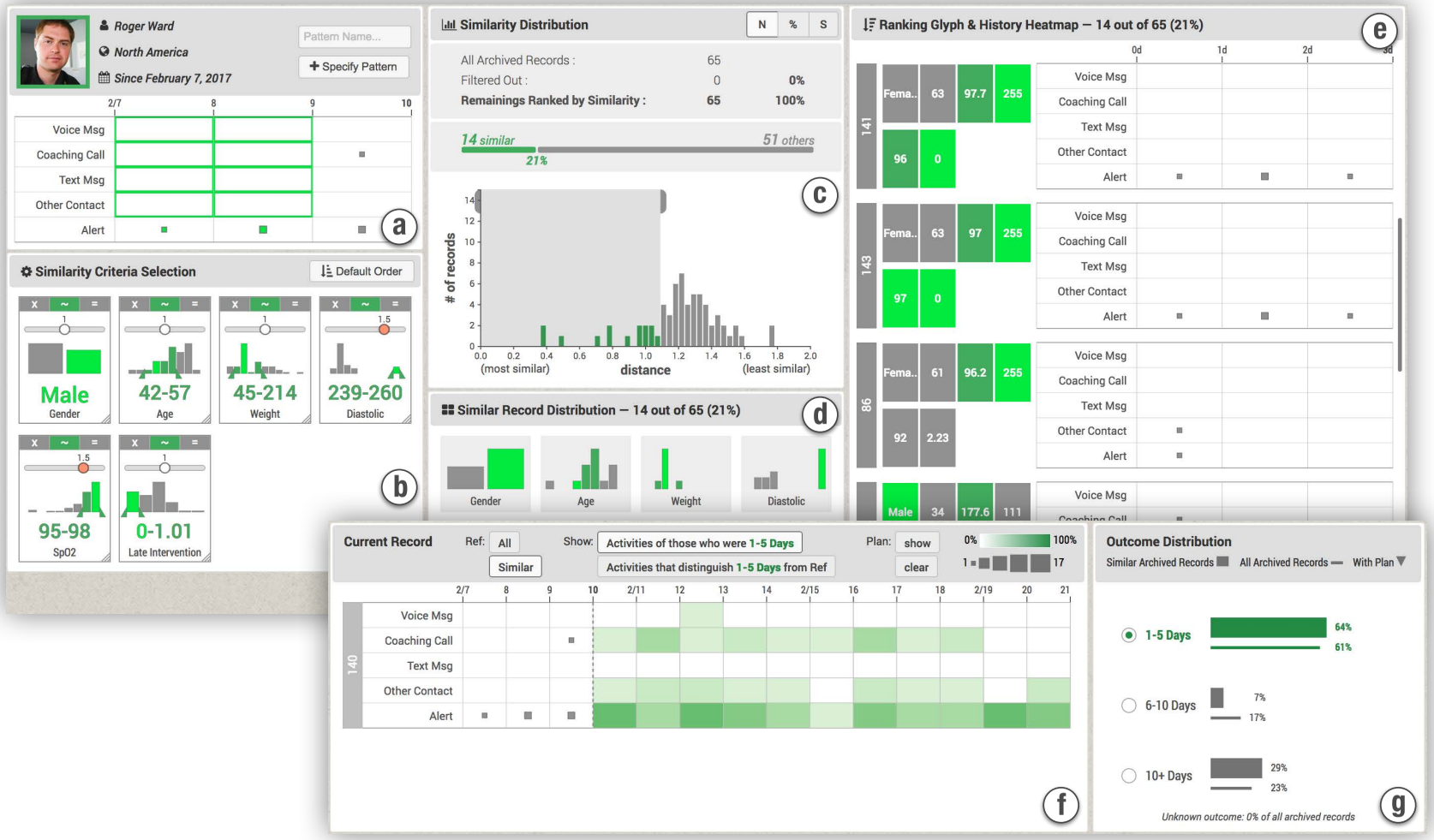

Figure 6.3: This figure illustrates a synthetic dataset of patient records. Event categories include voice message, coaching call, text message, other contact, and health alert (a). All available record attributes (gender, age, and recent readings) are used as similarity criteria by default (b). A new criterion is created to capture the temporal pattern of not being contacted by health coaches during the first two days of alerts (a). Three types of outcomes are defined based on the time spent on resolving the alerts (g). The top 21% most similar records are selected as the peer group (c).

6.3.2 Data

The research setting includes 107 health insurance plan members that were pragmatically 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 specials 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, birthday, weight), test results (diastolic, SpO2), outreaches (1,004 events including coaching call, voice message, text message, and others), and care gaps (2,626 alert events).

6.3.3 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 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%.

6.3.4 Feedback

Reviewing EventAction with health analysts provided actionable insights. A powerful component of EventAction is that it allows for hypothesis testing of patient results. For example, a health coach (or care manager) can pose the question to the data: "What could happen when similar patients to the patient under inquiry did X?" Further, the system allows for addressing population health strategies through easily identifying and segmenting patient cohorts by customizable data parameters.

The study indicated that more interpretation of results was needed. I anticipate this will be built into training materials and more tooltips will be included that 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 raises privacy concerns that one may be exposing peers unnecessarily. Additional work to conceive appropriate anonymization for large-scale implementation is needed. Finally, the heatmap meanings were not totally clear at first, although the color darkness made it easy to see where similar patients achieved desired results.

I recognize the limitation that the results are for one mHealth care management system and involve feedback from a limited number of health analysts, however, this exploratory case study provides a novel visualization with innovative and insightful findings for future exploration.

6.4 Incomplete Case Studies

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

6.4.1 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 redirected to find appropriate strategies to aggregate the event categories and to change the method the data is recorded (e.g., asking the operators to select from a list of possible event categories).

6.4.2 Too Complex Temporal Patterns

One healthcare case study was incomplete due to the extreme complexity of the temporal patterns. 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 can 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 became very complex and unique and spanned over years, making it was difficult to visually confirm the common patterns between the similar records and the seed record or to assess their similarities. The study partner decided to simplify the dataset before continuing the analysis, such as extracting events with a time window and coalescing hidden complex events into one [54].

6.4.3 No Suitable Outcome

Another healthcare case study was incomplete because the study partners were unable to define a suitable outcome for the records. 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 can recommend possible treatment plans for a current patient by finding similar previous patients. After a few visits and meetings, I was 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 for the outcomes of the patients (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 and returned to gather the missing outcome information.

6.5 Summary

This chapter has reported on four case studies that illustrate the use of EventAction in three application domains: education, marketing, and healthcare. The case studies were conducted with real users and using real-world datasets, providing evidence of the effectiveness of generating event sequence recommendations based on personal histories. I have also described three other case studies that were not completed for a variety of reasons to help potential users identify conditions for suitable applications of EventAction.

Chapter 7: Discussion and Future Directions

Recommender systems are being 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 dissertation 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. 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.

In this final chapter, I first describe the design guidelines and usage guidelines produced through my studies. Then, I summarize the contributions this dissertation has made toward explainable event sequence recommendations. Next, I discuss several promising future directions to extend my current software prototypes and

studies. Finally, I conclude the dissertation with some high-level closing remarks.

7.1 Guidelines

In this section, I describe five major 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 (Table 7.1). These guidelines are produced through my empirical studies of interface components and case studies in three domains, including education, marketing, and healthcare.

G1: Center the interface design on the seed record.
G2: Increase algorithm transparency with visualizations and user controls.
G3: Show both individual-level details and group-level overviews.
G4: Include both record attributes and temporal activities.
G5: Support flexible analytical workflows and satisfy different users' needs.
G6: Use rich, large, and representative data.
G7: Remind users that it is okay to be unique among past paths.
G8: Encourage collaborative use with an experienced advisor.

Table 7.1: Design guidelines (G1-5) for the construction of event sequence recommendation user interfaces and usage guidelines (G6-8) for mitigating the ethical issues in dealing with personal histories.

7.1.1 Design Guidelines

G1. Center the interface design on the seed record. Unlike many other event sequence visualization tools [7, 52, 53, 55], the analytical workflow of EventAction is oriented by a seed record. Centering the interface design on the seed record can emphasize the workflow and keep users focused on the tasks of finding similar records and making action plans for the seed record.

For example, when designing the LikeMeDonuts, I 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. My 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. In addition, how people perceive the similarity between personal records is very subjective depending on their preferences, experiences, and beliefs, and has been dismissed by some as a slippery notion [75]. 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. Notably, this dissertation mainly focused on the scenario of making critical life decisions when users demand more controls and context even at the cost of added complexity [70, 71]. My designs and findings may not be applicable to recommender systems for making less critical decisions in entertainment and shopping applications.

G3. Show both individual-level details and group-level overviews. Review-

ing 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 size of the group, 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, I 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 and satisfy different users' needs. I have noticed many different user workflows in user studies and case studies, deviating from the default analytical workflow (Figure 5.10). For example, some 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 analyses, and specific datasets and analytical goals. 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.

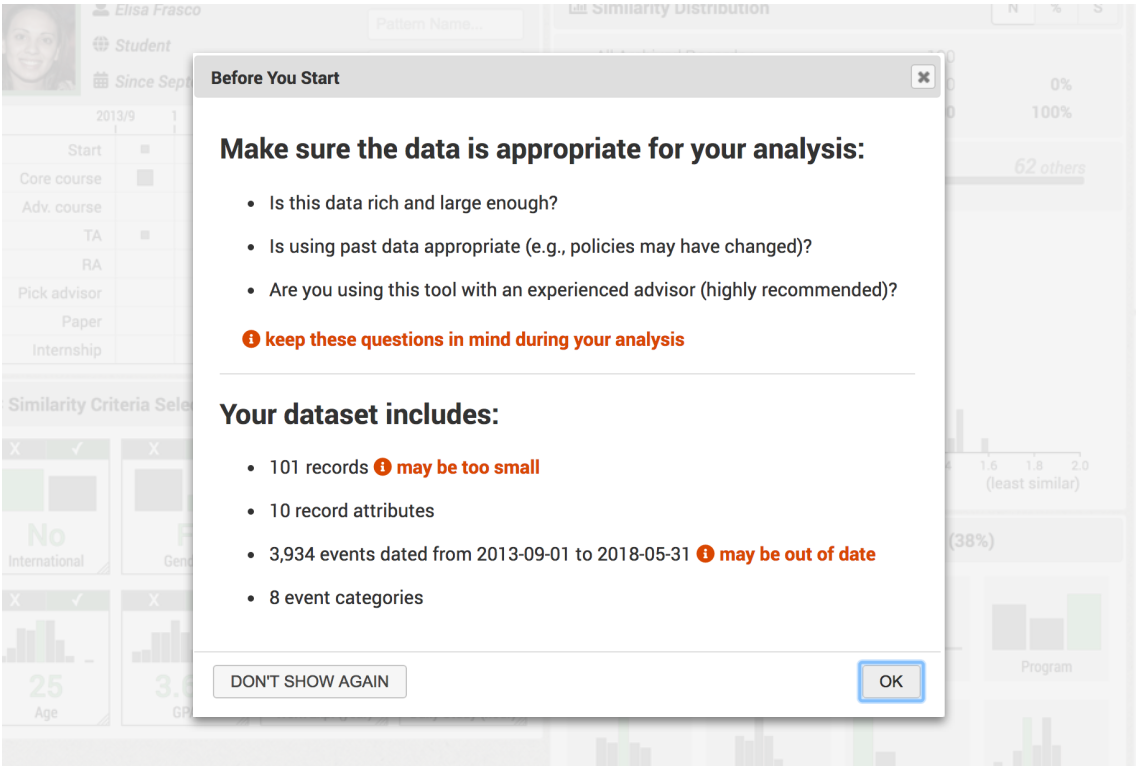

Figure 7.1: The startup screen of EventAction that prompts usage guidelines and identifies potential issues and biases in the data.

7.1.2 Ethical Issues and Usage Guidelines

Reviewing ethical issues is important in dealing with personal histories. I discuss the ethical issues faced in my studies and describe three usage guidelines for mitigating those issues. EventAction provides a startup screen that prompts these usage guidelines and identifies potential issues and biases in the data (Figure 7.1).

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 quantity and quality of the data available. To better profile the current advisee 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 data available do not represent people adequately and there are few similar records exist, or when there are errors or missing attributes in the data [94]. 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 remain 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 are 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 expe-

rienced 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 [79]. EventAction mitigates this issue by giving transparent data access to both advisors and advisees and involving them in the decision-making process.

7.2 Summary of Contributions

This dissertation contributes an analytical workflow, an interactive system, and design guidelines identified in empirical studies and case studies, opening new avenues of research in explainable event sequence recommendations based on personal histories. It enables people to make better decisions for critical life choices with higher confidence. The concrete contributions of this dissertation are:

- **A systematic analytical workflow for event sequence recommendation that will be applicable in diverse applications.** The workflow was developed and refined based on my observations of user behaviors during empirical studies and case studies. The typical workflow starts from a seed record and the first step is to find a group of similar archived records. After submitting the similar records and the reward function, 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, adjusting the reward function to receive an updated recommendation, or refining the similar records. Many small devia-

tions have also been observed in the studies, such as skipping or changing the order of certain steps. 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.

- **An interactive prescriptive analytics system and user interfaces to assist users in making action plans and to raise users' confidence in the action plans, and the integration of an automatic sequence recommendation algorithm to reduce users' effort in using the system.** Through iterative usability studies and case studies, I have designed, developed, and refined two user interfaces: PeerFinder and EventAction. PeerFinder presents a visual interface that enables users to find and explore records that are similar to a seed record. To encourage engagement and inspire users' trust in the results, PeerFinder provides different levels of controls and context that allow users to adjust the similarity criteria. EventAction provides a visual analytics approach to (1) identify similar records, (2) explore potential outcomes, (3) review recommended event sequences that might help achieve the users' goals, and (4) interactively assist users as they define a personalized action plan associated with a probability of success. The final EventAction system integrates the PeerFinder visual components and an automatic sequence recommendation algorithm, supporting a seamless analytical workflow

for developing action plans to achieve users' desired outcomes.

- **Empirical studies of interface components and case studies in three domains that provide evidence of the effectiveness of generating event sequence recommendations based on personal histories.** The studies include (1) a student advising case study conducted with a professor on a dataset of real students' data, (2) two digital marketing case studies conducted with 5 marketing analysts and using real-world datasets of customer records, and (3) a medical case study conducted with two health analysts using real-world patient health records.

- **Design guidelines for the construction of event sequence recommendation user interfaces and usage guidelines for mitigating the ethical issues in dealing with personal histories.** The design guidelines include (1) center the interface design on the seed record, (2) increase algorithm transparency with visualizations and user controls, (3) show both individual-level details and group-level overviews, (4) include both record attributes and temporal activities, and (5) support flexible analytical workflows and satisfy different users' needs. The usage guidelines include (1) use rich, large, and representative data, (2) remind users that it is okay to be unique among past paths, and (3) collaborative use with an experienced advisor.

7.3 Future Directions

The goal of this dissertation was to explore the research area of explainable event sequence recommendations and open up new directions for future researchers. In addition to the power of the user controls and visualizations provided by EventAction, and the contributions of this dissertation, I have identified huge opportunities to extend my current software prototypes and studies. In this section, I discuss various ways in which this work can be further developed.

7.3.1 Scaling Up

Scalability becomes an issue for most interactive visualizations as the size of the data grows. A larger number of archived records can slow down the computation of similarity, the rendering of the visualizations, and users' interpretation of the results. While using powerful machines can accelerate the computation and rendering, how to reduce human effort in analyzing larger datasets remains challenging for EventAction. I propose three future directions to support analyses of extremely large datasets, such as millions of online customer records.

7.3.1.1 Seed Group

While making action plans for a 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), for example, customers in Maryland who have received

at least three promotional emails but have not purchased yet. Then, marketers can explore archived customers similar to the seed group and develop campaign strategies to increase the purchase rate of the group. A direct and promising solution is to allow users to specify a query characterizing the seed group using EventAction's seed record timeline and similarity criteria controls, with additional visualizations and controls for reviewing and refining the seed group.

7.3.1.2 Record Categorization

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. For example, Vadrevu et al. [112] proposed a clustering method for news search results using a composite metric of meta data, textual data, and query terms, and Chakrabarti et al. [113] proposed a categorization approach by building a hierarchy based on data attributes. However, few papers addressed this problem in the context of event sequences. One potential solution is to categorize the similar records by their differences compared to the seed record, such as missing or having extra events, or variations in the ordering or temporal distribution of events. Users can easily understand how the records in each category deviate from the seed record and explore a category of records at a time.

7.3.1.3 Criteria Selection

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 [89, 90]. Automatic techniques for assigning weights to the criteria or classifying the criteria into multiple groups may also be useful (e.g., one for demographics, another for academic experience, and a third for work experience), but formal evaluation is needed to identify and quantify benefits.

7.3.2 Supporting Collaboration

The typical usage scenario of EventAction is an advisor collaborating with an advisee on personalizing an action plan for the advisee. In this scenario, both the advisor and advisee are involved in exploring the data and creating the plan. The degree of engagement will influence the quality of the plan and advisee's trust in the plan. Unlike existing collaborative visualization systems as summarized by Isenberg et al. [114], 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. Currently, EventAction treats advisors as the main users, who control the system, explore the visualizations, and describe the findings to advisees. 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.

7.3.3 Celebrating Diversity

Beyond similarities and differences, visualization tools can also be designed to guide the creation of diverse teams. Diversity can drive innovation in teams [95]. 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 diverse 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.

7.4 Closing Remarks

This chapter summarized all results and contributions from this dissertation and discussed promising future opportunities to extend the current software prototypes and studies. The contributions of this dissertation have been implemented in EventAction, an interactive prescriptive analytics system along with a system-

atic analytical workflow, to assist users in making action plans and to raise users' confidence in the action plans. Empirical studies in three domains have provided evidence of the effectiveness of generating event sequence recommendations based on personal histories. Through the design, implementation, and evaluation of EventAction, I 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. I believe this dissertation will open new avenues of research in explainable event sequence recommendations based on personal histories and enable people to make better decisions for critical life choices with higher confidence.

Bibliography

[1] Emanuel Zgraggen, Steven M. Drucker, Danyel Fisher, and Robert DeLine. (s|qu)eries: Visual regular expressions for querying and exploring event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2683–2692, 2015.

[2] Josua Krause, Adam Perer, and Harry Stavropoulos. Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):91–100, 2016.

[3] Megan Monroe, Rongjian Lan, Juan Morales del Olmo, Ben Shneiderman, Catherine Plaisant, and Jeff Millstein. 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*, pages 2349–2358, 2013.

[4] Catherine Plaisant, Rich Mushlin, Aaron Snyder, Jia Li, Daniel Heller, and Ben Shneiderman. LifeLines: Using visualization to enhance navigation and analysis of patient records. In *Proceedings of the AMIA Symposium*, pages 76–80, 1998.

[5] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. LifeLines: Visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 221–227, 1996.

[6] Taowei David Wang, Catherine Plaisant, Alexander J Quinn, Roman Stanchak, Shawn Murphy, and Ben Shneiderman. Aligning temporal data by sentinel events: Discovering patterns in electronic health records. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 457–466, 2008.

[7] Krist Wongsuphasawat and David Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2659–2668, 2012.

[8] Lyndsey Franklin, Catherine Plaisant, Kazi Minhazur Rahman, and Ben Shneiderman. Treatmentexplorer: An interactive decision aid for medical risk communication and treatment exploration. *Interacting with Computers*, 28(3):238–252, 2016.

[9] Theresia Gschwandtner, Wolfgang Aigner, Katharina Kaiser, Silvia Miksch, and Andreas Seyfang. Carecruiser: Exploring and visualizing plans, events, and effects interactively. In *IEEE Pacific Visualization Symposium*, pages 43–50, 2011.

[10] Sana Malik, Ben Shneiderman, Fan Du, Catherine Plaisant, and Margret Bjarnadottir. High-volume hypothesis testing: Systematic exploration of event sequence comparisons. *ACM Transactions on Interactive Intelligent Systems*, 6(1):9:1–9:23, 2016.

[11] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. MatrixWave: Visual comparison of event sequence data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 259–268, 2015.

[12] Alexandra Papoutsaki, Hua Guo, Danae Metaxa-Kakavouli, Connor Gramazio, Jeff Rasley, Wenting Xie, Guan Wang, and Jeff Huang. Crowdsourcing from scratch: A pragmatic experiment in data collection by novice requesters. In *AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[13] Richard Snodgrass. The temporal query language tquel. *ACM Transactions on Database Systems*, 12(2):247–298, 1987.

[14] Fabio Grandi. T-sparql: A tsql2-like temporal query language for rdf. In *In International Workshop on Querying Graph Structured Data*, pages 21–30, 2010.

[15] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

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

[17] Liren Chen and Katia Sycara. Webmate: A personal agent for browsing and searching. In *Proceedings of the International Conference on Autonomous Agents*, pages 132–139, 1998.

[18] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[19] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[20] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the International Conference on World Wide Web*, pages 285–295, 2001.

[21] Bruce Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–46, 1997.

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

[23] Francesco Ricci, Dario Cavada, Nader Mirzadeh, and Adriano Venturini. Case-based travel recommendations. *Destination Recommendation Systems: Behavioural Foundations and Applications*, pages 67–93, 2006.

[24] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, volume 60, 1999.

[25] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In *Web Mining: From Web to Semantic Web*, pages 57–76. 2004.

[26] Abdul Majid, Ling Chen, Gencai Chen, Hamid Turab Mirza, Ibrar Hussain, and John Woodward. A context-aware personalized travel recommendation system based on geotagged social media data mining. *International Journal of Geographical Information Science*, 27(4):662–684, 2013.

[27] Augusto Q Macedo, Leandro B Marinho, and Rodrygo LT Santos. Context-aware event recommendation in event-based social networks. In *Proceedings of the ACM Conference on Recommender Systems*, pages 123–130, 2015.

[28] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

[29] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 210–217, 1995.

[30] Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. On the recommending of citations for research papers. In *Proceedings of the ACM conference on Computer supported cooperative work*, pages 116–125, 2002.

[31] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 830–831, 2002.

[32] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4):13:1–13:19, 2016.

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

[34] 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. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[35] Weizhong Li and Adam Godzik. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[36] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[37] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *Proceedings of the Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[38] Simone Santini and Ramesh Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[39] Paul Wicks, Michael Massagli, Jeana Frost, Catherine Brownstein, Sally Okun, Timothy Vaughan, Richard Bradley, and James Heywood. Sharing health data for better outcomes on PatientsLikeMe. *Journal of Medical Internet Research*, 12(2):e19, 2010.

[40] Melanie Swan. Crowdsourced health research studies: An important emerging complement to clinical trials in the public health research ecosystem. *Journal of Medical Internet Research*, 14(2):e46, 2012.

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

[42] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. 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*, pages 678–684, 2005.

[43] Ashish Sureka and Pranav Prabhakar Mirajkar. An empirical study on the effect of different similarity measures on user-based collaborative filtering algorithms. In *Pacific Rim International Conference on Artificial Intelligence*, pages 1065–1070, 2008.

[44] Heikki Mannila and Pirjo Ronkainen. Similarity of event sequences. *TIME*, 97:136–140, 1997.

[45] Krist Wongsuphasawat and Ben Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 27–34, 2009.

[46] Katerina Vrotsou and Camilla Forsell. A qualitative study of similarity measures in event-based data. In *Symposium on Human Interface*, pages 170–179, 2011.

[47] Katerina Vrotsou, Anders Ynnerman, and Matthew Cooper. Are we what we do? exploring group behaviour through user-defined event-sequence similarity. *Information Visualization*, pages 232–247, 2013.

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

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

[50] Beverly L Harrison, Russell Owen, and Ronald M Baecker. Timelines: An interactive system for the collection and visualization of temporal data. In *Graphics Interface*, pages 141–141, 1994.

[51] Gerald M Karam. Visualization using timelines. In *Proceedings of the ACM International Symposium on Software Testing and Analysis*, pages 125–137, 1994.

[52] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. LifeFlow: Visualizing an overview of event sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1747–1756, 2011.

[53] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, 2013.

[54] Fan Du, Ben Shneiderman, Catherine Plaisant, Sana Malik, and Adam Perer. Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1636–1649, 2017.

[55] David Gotz and Harry Stavropoulos. DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783–1792, 2014.

[56] Srivatsan Laxman and P Shanti Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

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

[58] Heikki Mannila, Hannu Toivonen, and A Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3):259–289, 1997.

[59] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on knowledge and data engineering*, 16(11):1424–1440, 2004.

[60] Mohammed J Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.

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

[62] Adam Perer and Fei Wang. Frequence: interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 153–162, 2014.

[63] Adam Perer and David Gotz. Data-driven exploration of care plans for patients. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 439–444, 2013.

[64] Richard O Mason. Four ethical issues of the information age. *MIS Quarterly*, pages 5–12, 1986.

[65] Helen Nissenbaum. Computing and accountability. *Communications of the ACM*, 37(1):72–81, 1994.

[66] Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Transactions on Information Systems*, 14(3):330–347, 1996.

[67] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. EventAction: Visual analytics for temporal event sequence recommendation. In *Proceedings of the IEEE Visual Analytics Science and Technology*, pages 61–70, 2016.

[68] Christopher A Longhurst, Robert A Harrington, and Nigam H Shah. A 'Green Button' for using aggregate patient data at the point of care. *Health Affairs*, 33(7):1229–1235, 2014.

[69] Sandy H Huang, Paea LePendu, Srinivasan V Iyer, Ming Tai-Seale, David Carrell, and Nigam H Shah. Toward personalizing treatment for depression: Predicting diagnosis and severity. *Journal of the American Medical Informatics Association*, 21(6):1069–1075, 2014.

[70] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 5498–5509, 2017.

[71] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.

[72] Jürgen Koenemann and Nicholas J Belkin. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 205–212, 1996.

[73] Jaime Teevan, Christine Alvarado, Mark S Ackerman, and David R Karger. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 415–422, 2004.

[74] World Health Organization. The tenth revision of the international classification of diseases and related health problems (ICD-10), 1992.

[75] Lieven Decock and Igor Douven. Similarity after goodman. *Review of Philosophy and Psychology*, 2(1):61–75, 2011.

[76] Jan De Leeuw and Sandra Pruzansky. A new computational method to fit the weighted euclidean distance model. *Psychometrika*, 43(4):479–490, 1978.

[77] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[78] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2017.

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

[80] Carlos A Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4):13:1–13:19, 2016.

[81] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. Visual interfaces for recommendation systems: Finding similar and dissimilar peers. *ACM Transactions on Intelligent Systems and Technology*, pages 1–23, 2018.

[82] Michael Glueck, Peter Hamilton, Fanny Chevalier, Simon Breslav, Azam Khan, Daniel Wigdor, and Michael Brudno. PhenoBlocks: Phenotype comparison visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):101–110, 2016.

[83] Jeffrey Heer and George Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[84] Fanny Chevalier, Nathalie Henry Riche, Catherine Plaisant, Amira Chalbi, and Christophe Hurter. Animations 25 years later: New roles and opportunities. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 280–287, 2016.

[85] Junpeng Wang, Xiaotong Liu, Han-Wei Shen, and Guang Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):81–90, 2017.

[86] Geoffrey M Draper, Yarden Livnat, and Richard F Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):759–776, 2009.

[87] Benjamin B Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.

[88] Catherine Plaisant, Johnny Wu, A Zach Hettinger, Seth Powsner, and Ben Shneiderman. Novel user interface design for medication reconciliation: An evaluation of Twinlist. *Journal of the American Medical Informatics Association*, 22(2):340–349, 2015.

[89] Matthew Louis Mauriello, Ben Shneiderman, Fan Du, Sana Malik, and Catherine Plaisant. Simplifying overviews of temporal event sequences. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 2217–2224, 2016.

[90] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2648–2659, 2017.

[91] Yanhong Wu, Naveen Pitipornvivat, Jian Zhao, Sixiao Yang, Guowei Huang, and Huamin Qu. egoslider: Visual analysis of egocentric network evolution. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):260–269, 2016.

[92] Nan Cao, Yu-Ru Lin, Fan Du, and Dashun Wang. Episogram: Visual summarization of egocentric social interactions. *IEEE Computer Graphics and Applications*, 36(5):72–81, 2016.

[93] Jian Zhao, Michael Glueck, Fanny Chevalier, Yanhong Wu, and Azam Khan. Egocentric analysis of dynamic networks with egolines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 5003–5014, 2016.

[94] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, 2016.

[95] Sylvia Ann Hewlett, Melinda Marshall, and Laura Sherbin. How diversity can drive innovation. *Harvard Business Review*, 91(12):30–30, 2013.

[96] James R Evans and Carl H Lindner. Business analytics: The next frontier for decision sciences. *Decision Line*, 43(2):4–6, 2012.

[97] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. Springer, 2011.

[98] J Donaldson. Music recommendation mapping and interface based on structural network entropy. In *IEEE International Conference on Data Engineering Workshop*, pages 811–817, 2007.

[99] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[100] Taowei David Wang, Catherine Plaisant, Ben Shneiderman, Neil Spring, David Roseman, Greg Marchand, Vikramjit Mukherjee, and Mark Smith. Temporal summaries: Supporting temporal categorical searching, aggregation and

comparison. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1049–1056, 2009.

[101] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 483–492, 2013.

[102] Guy Shani, Ronen I Brafman, and David Heckerman. An mdp-based recommender system. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 453–460, 2002.

[103] Fan Du, Sana Malik, Georgios Theocharous, and Eunyee Koh. Personalizable and interactive sequence recommender system. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 1–6, 2018.

[104] Douglas J White. Real applications of Markov decision processes. *Interfaces*, 15(6):73–83, 1985.

[105] Georgios Theocharous, Nikos Vlassis, and Zheng Wen. An interactive points of interest guidance system. In *Proceedings of the International Conference on Intelligent User Interfaces Companion*, pages 49–52, 2017.

[106] Alexis Gabadinho and Gilbert Ritschard. Analyzing state sequences with probabilistic suffix trees: The PST R package. *Journal of Statistical Software*, 72(3):1–39, 2016.

[107] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

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

[109] Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized markov decision processes. In *Conference on Learning Theory*, pages 861–898, 2015.

[110] Ben Shneiderman and Catherine Plaisant. 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*, pages 1–7, 2006.

[111] Fan Du, Sana Malik, Eunyee Koh, and Georgios Theocharous. Interactive campaign planning for marketing analysts. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 1–6, 2018.

[112] Srinivas Vadrevu, Choon Hui Teo, Suju Rajan, Kunal Punera, Byron Dom, Alexander J Smola, Yi Chang, and Zhaohui Zheng. Scalable clustering of news search results. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 675–684, 2011.

[113] Kaushik Chakrabarti, Surajit Chaudhuri, and Seung-won Hwang. Automatic categorization of query results. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 755–766, 2004.

[114] Petra Isenberg, Niklas Elmqvist, Jean Scholtz, Daniel Cernea, Kwan-Liu Ma, and Hans Hagen. Collaborative visualization: definition, challenges, and research agenda. *Information Visualization*, 10(4):310–326, 2011.