EventScope: Finding Discriminating Event Patterns Based on Different Outcomes

CMSC734 Final Project Report, Dec 2013

Andres Arellano  
Dept. of Computer Science  
University of Maryland,  
College Park  
andres.arellano@gmail.com  

Michael Sorensen  
Dept. of Computer Science  
University of Maryland,  
College Park  
mdsoren@umd.edu  

Gregory Kramida  
Dept. of Computer Science  
University of Maryland,  
College Park  
gkramida@cs.umd.edu  

Shobeir Fakhraei  
Dept. of Computer Science  
University of Maryland,  
College Park  
shobeir@cs.umd.edu  

Hao Li  
Dept. of Computer Science  
University of Maryland,  
College Park  
haoli@cs.umd.edu  

ABSTRACT

Discovering temporal patterns in large datasets is a challenging problem because researchers must explore a huge space of possible correlations. EventScope is a visualization and ranking tool to assist in identification of temporal patterns leading to, or following, different events. In order to identify non-trivial patterns, EventScope splits the data based on user-specified reference events. Patterns are sorted based on differences found between these two groups. By providing information regarding an event of interest relative to two outcomes, more useful temporal patterns can be identified. EventScope was evaluated by conducting usability tests with events derived from one season of Chicago Bulls basketball games. Usability test results from non-expert users were generally positive and users were able to successfully identify several strong relations.

General Terms

H.5.2 [Information Interfaces and Presentation]: User Interfaces, Graphical user interfaces (GUI)

Keywords

Information Visualization, Temporal Event Sequence

1. INTRODUCTION

Understanding correlation of temporal events in large datasets is an important problem in a number of domains, including: medicine, sports commentary, and marketing. EventScope focuses on finding pairs of events that exhibit strong temporal relations. For example, in the medical domain, understanding that taking a drug is followed by a favorable outcome is very useful for generating hypotheses. In large datasets with multiple types of events, identifying temporal patterns is very difficult. Handling a large number of event combinations, measuring strength of the relation and ranking potential relationships are several problem areas that make this task complex.

A large number of event types makes the number of potential temporal patterns exponentially large. If we have \( E \) types of events in the dataset, the number of possible correlations is \( E^2 \). In general, naively examining temporal patterns of length \( N \) requires considering \( E^N \) combinations of events. To examine a sequence of events to determine the strength of a temporal pattern, each element in the sequence must be examined. If the average sequence length is \( S \), the total amount of computation for a naive temporal sequence mining is roughly \( O(E^N \times S) \). The process grows exponentially with the number of types of events and the length of the pattern.

In addition to scaling issues, temporal sequence mining algorithms face challenges in identifying strong patterns. Measuring the strength of a pattern is dependent upon the dataset and the questions to be answered by the interface. Numerous scoring techniques have been proposed, however algorithmically determining the best scoring heuristic for the dataset and desired outcome remains an unsolved problem.

EventScope is a visualization tool to assist in the identifying temporal event relations. EventScope varies on the common temporal event mining processes by considering outcomes or labels. Simply, EventScope aims to highlight patterns that distinguish between two user specified groups. With EventScope, users can find potentially find pattern of the following type; event \( A \) occurs followed by event \( B \), and event \( A \) follows event \( B \). Using different discriminating measures to rank patterns EventScope can highlight the most informative patterns. The addition of the outcome or label information to
the pattern mining and its adoption in a visualization tool is introduced in EventScope.

EventScope splits data into two groups based on a user selected binary attribute, and identifies pair association rules that vary among the two portions of the split dataset. In this setting patterns that are most different between two groups are assumed to be most interesting. It is expected that relations that occur in both sets are uninteresting.

For example, consider a dataset where basketball events are recorded and one is interested in identifying events that lead to a successful 3-point attempt. The EventScope process will begin by splitting the data into two groups: successful and unsuccessful 3-point attempts. Inspired by measures proposed in PairFinder [3], patterns will be scored so that the patterns that can discriminate between the two populations score highly. High scoring patterns are candidates that can be reviewed by experts to determine if they cause a 3-point attempt successful.

Visually, EventScope’s temporal potential patterns are displayed in a sorted order. Temporal patterns are displayed in small charts lacking labels in order to enable efficient scanning for patterns of interest. While ranking relations remains important, the ability to scan multiple relations quickly relaxes the requirement.

This paper only considers patterns consisting of pairs of events. These patterns have the form; event A frequently precedes the reference event B in this split of the population, but not the other split of the population.

Limiting EventScope to identifying pairs of events limits the number of possible relations to consider and makes the visualization less cluttered. Additionally, pair temporal relation mining is a base case for multiple event relation mining. Techniques must work on pairs of events before they can be extended to find longer patterns. Furthermore, if there is a longer discriminating pattern of events, at least two events in the pattern should be different which should be captured by pair relation mining.

We have developed a prototype of the EventScope tool to determine the effectiveness of the EventScope interface and temporal association mining algorithms by usability studies. In the study, we loaded a basketball dataset to the prototype and asked users to find events that lead to a successful or unsuccessful 3-point shot. Test subjects were not data analyst and tended to be unfamiliar with temporal event mining software beyond a basic Office Suite. Most usability study participants were able to successfully identify that successful 3-point shots lead to a change in possession.

2. RELATED WORK

There is extensive literature devoted to finding sequence patterns and temporal patterns in large datasets. A combination of effective heuristics, statistics and visualizations has produced several useful tools including LifeLines [11], EventFlow [7], and PairFinder [3] that have been applied to sports, medical and business marketing applications.

2.1 Mining Temporal Sequences

Sequence mining is the problem of finding an ordered pattern of events that occur together in a set of data. For example, a sequence mining algorithm might determine that event A, event B and event C commonly occur together, with that order. The Apriori algorithm [1] was groundbreaking preliminary work in this area, providing an ability to efficiently identify long sequences of events in large datasets. Apriori starts by finding short sequences of events that occur most frequently in the dataset and exceed a user provided support threshold. Building on the shorter sequences, longer sequences of events are identified.

A related problem to mining sequences is temporal event sequence mining, where in addition to identifying sequences of events, the timing of events is also considered. Commonly, temporal patterns heuristics are developed based on some properties that are identified by users as being interesting. Lallich et al. [5] discusses the effectiveness of several ranking approaches including: BayesFactor, Lift, and Pearson’s correlation coefficient. Alternatively, temporal data can be reduced to sequence data and mined using sequence mining techniques.

Statistical techniques commonly used in sequence mining are geared towards identifying commonly co-occurring events. A determination of causality is left to the user. Inferring causality from data is well studied. Concepts borrowed from the scientific process and experimental design have become increasingly important [12]. Roughly the process includes, dividing a population into two groups, treating one group, measuring outcomes and testing a hypothesis for statistical significance. Additionally, Bayesian networks [10] have been developed to test causality hypothesis based on data. The addition of experimental design concepts and rigorous statistical techniques to add a causality indicator into a visualization tool would offer the user a powerful way to explore their data.

2.2 Visualization Tools

In addition to the ranking approaches for temporal patterns, a number of successful visualization tools have been developed. EventFlow [7] is a visualization tool that aligns event times within a record to a single reference event. Event records are then sorted based on the most frequently occurring event following the reference event. This technique enabled efficient identification of most common patterns and common overall trends, however identification of less frequently occurring patterns is also useful. An interesting capability of Eventflow is handling interval events as well as point events.

PairFinder [3] identifies strong relations between two events and uses several interesting heuristics to rank the patterns. The heuristics are proven to extract strong patterns, however many strong patterns are obvious. For example, in a basketball dataset one might find that the first quarter always begins 15 minutes before the second quarter. Unless there was a data acquisition error, this relation would occur in all of the data. For an arbitrary dataset, the number of obvious rules that are identified may become burdensome to a user and should be minimized.

The TimelineTrees [2] visualization tool introduced a tree view of temporal events. Tree views can be difficult to see if
the number events or unique patterns is very large. Outflow [16] is a medical record specific visualization tool that produces a flow graph that shows the most commonly occurring patterns. Outflow incorporates patient outcome by coloring edges on the flow graph, with red indicating negative outcomes. Outflow relies on user visual identification of strong patterns.

EventScope extends these approaches with additional contributions aimed at improving the interface to visualize these datasets, offering more control over data manipulation, and adding outcome information to temporal event mining. In contrast with EventFlow, to control the scope of the project EventScope only considered point events.

3. OUR METHOD

Given a dataset with multiple independent timed event sequences, there is a variety of questions an analyst may ask. One simple question is “for a particular event type, which events are more likely to cause it or be caused by it?”. PairFinder [3] provides assistance specifically in this problem domain. Another kind of questions is “given a particular outcome, what is the most likely sequence that would lead to this outcome?”. This question is addressed by LifeLines [11] and [7] applications. There question one can ask is “provided two different results, which event(s) lead to one, but not to the other? And, conversely, given two kinds of events, which event(s) only follow one but not the other?”. EventScope aims to address these questions in the third category.

3.1 Problem Statement

The problem that EventScope addresses is formulated as the following: Given any set of event sequences or records $S = \{ s_1, s_2, \ldots, s_n \}$, composed of events from the following set:

$$e_{all} = \{ e_a, e_b, e_1, e_2, \ldots, e_n \}$$ (1)

rank events based on which event is (i) most likely to proceed outcome event $e_b$ but not $e_a$ (and vice versa) or (ii) most likely to follow $e_a$ but not $e_b$ and vice versa. It is not imperative for $e_a$ and $e_b$ to be mutually exclusive, i.e. a single record can include both. However, for most applications $e_a$ and $e_b$ are mutually exclusive.

We propose a simple framework to rank such distributions of point events relative to two distinct outcome events. This framework combines several components indicating the amount of possible “discriminativeness” for distributions of timing in events in sets $e_a$ and $e_b$. Each one of these components relies on some measure that can potentially serve as an indicator of causality for the two outcome events. We designate the value of a particular indicator $j$ for event $i$ causing outcome event $x$ as

$$I_j(i, x) = f_j(e_i, e_x).$$ (2)

where $f_j$ is some black-box function suggesting causality (or correlation) strength.

The basic measure of discriminativeness we define for the $i$-th event using indicator $j$ is the following:

$$d_{ij} = |I_j(e_i, e_a) - I_j(e_i, e_b)|$$ (3)

Since we are dealing with records and events from multiple domains, the appropriateness of any indicator in determining the discriminativeness may vary from one domain to another or even from one reference event type to the other, we offer a user defined binary weigh for each discriminative measure, $\lambda_j$. The overall discriminativeness of the events is computed as:

$$D_i = \sum_j \lambda_j d_{ij}$$ (4)

3.2 Ranking Measures

The Ranking Measures in EventScope is inspired by several proposed interestness measures in PairFinder [3]. It should be noted that any of the phenomena these indicators are based on are not protected against effects of confounding variables, i.e. if co-occurrence of the phenomenon with the outcome is caused by external event which influences both. The indicators are listed below.

3.2.1 Occurrence

Occurrence ratio is a simple measure of interestingness. If events occur before a reference event, but rarely after, the combination of events is of interest.

Occurrence ratio considers the relative number of events occurring before the reference event and after the reference event. Mathematically occurrence ratio is expressed as:

$$Oc(x) = 2 \times \frac{\max(n^-, n^+)}{n^- + n^+} - 1$$

where $n^-$ is the number of events occurring before the reference event, and $n^+$ is the number of events occurring after the reference event. The heuristic is normalized so that the range is $[0, 1]$.

If a given event frequently occurs before a reference event, but not after, the reference event may have caused the preceding events to stop. Similarly, if a given event frequently occurs after a reference event, but not before, the reference event may enable the following event.

3.2.2 Peaks

Peak ratio is similar to the occurrence metric (3.2.1), except that the locations of the peaks event counts are substituted for the locations of events relative to the reference event. The result is a heuristic that considers clumps of events in a similar time span, not simply events in isolation. Mathematically peak ratio is expressed as:

$$Pk(x) = 2 \times \frac{\max(p^-, p^+)}{p^- + p^+} - 1$$

where $p^-$ is the number of peaks occurring before the reference event, and $p^+$ is the number of peaks occurring after the reference event. The heuristic is normalized so that the range is $[0, 1]$.

Peaks were calculated using a sliding window technique presented in [9]. In this peak finding algorithm, a histogram is created and a peak was detected if the number of events in a previous $k$ bins, is greater than the number of events in the following $k$ bins by some threshold. $k$ is set to a small odd number such as 3 or 5.
In practice, Peaks produces event rankings that are very similar to the occurrence metric. However, the peaks metric may be useful in identifying relations where the number of events is roughly the same before and after the reference event but the shape of the the event distribution varies before and after the reference event.

### 3.2.3 Periodicity
Events that occur with a certain periodicity, are of interest. For example, in the medical domain prescription drugs should be taken in given intervals. The periodicity for a time series is simply the period. To compute the period, the Fourier Transform of the time series is computed. The frequency with the maximum amplitude is determined. The inverse of this frequency is the period. The range of the periodicity metric is the set of positive real numbers.

### 3.2.4 Standard Deviation
The standard deviation describes the shape of a probability distribution. In general, events that are distributed normally around a mean, have a lower standard deviation than events that do not follow a normal distribution. Events scoring high according to this metric will have irregular distributions. Events that are irregularly distributed are interesting and require further investigation.

The range of the standard deviation metric is set of positive real numbers.

### 3.3 Comparing Ranking Measures
EventScope’s Ranking Panel (Figure 1) sorts the pairs of temporal event relations based on their discriminative power. The sorted records consist of pairs of relations: \(E \Rightarrow R_1\) and \(E \Rightarrow R_2\).

The ranking measures have varying ranges. In order combine or compare the ranking measures they are first normalized. All ranking measures are normalized to produce values in the \([0,1]\) range using the following equation:

\[
x_{\text{normalized}} = \begin{cases} 
0 & \text{max}(X) = \min(X) \\
\frac{x - \min(X)}{\max(X) - \min(X)} & \text{else}
\end{cases}
\]

where \(x\) is a value produced by the ranking measure and \(X\) is the set of all values that the ranking measure produced for this dataset. As a result, every metric will have a value of 0, and every metric where \(\max(X) \neq \min(X)\) will have a value of 1.

Once ranking measures are normalized, the data is sorted. EventScope sorts the pairs of temporal relations by defining a sort key for the pairs of temporal relations and then sorting the pairs in descending order by the sort key. The sort key for a pair of temporal event relations is defined as the following:

\[
SortKey(x, y) = |RankMeasure(x) - RankMeasure(y)|
\]

Sorting according to these values (i.e., Sortkeys), pairs of temporal relations that both score high will be sorted towards the bottom of the list. When both patterns score highly, the pattern is not useful for differentiating between the temporal relation \(x\)’s outcome and temporal relation \(y\)’s outcome, so the pair is uninteresting. Pairs of temporal events that both score low will also be sorted towards the bottom of the list. Patterns that are not sufficiently strong should be ignored. Only the events where one temporal event scores highly and the other temporal event scores low, are ranked towards the top of the list.

### 3.4 User Interface
In EventScope, we borrow from PairFinder the concept of aligning events to a reference event and then produce a histogram. EventScope differentiates itself from PairFinder by using the co-occurrences of two reference event classes with a third event, aligned by that third event.

The EventScope interface user interface was built in order to support the idea described above. It has three main parts: the Control Panel on the upper right side, the Ranking Panel on the left side and the Details Panel, on the bottom right side. The 3 panels and their positioning are shown in Figure 1.

Sections 3.4.1, 3.4.2 and 3.4.3 explain in detail each one of these parts.

#### 3.4.1 Control Panel
The control panel is divided in three rows, as we can see in Figure 2.

The top most row of the control panel allows users to select the dataset from a drop down list. Selecting a dataset is the required first action, as is indicated in the starting message shown in Figure 3. After that, the user needs to select the two reference events that will be used to compare the outcomes (e.g. Chicago Missed Shot-3 and Chicago Made Shot-3).

The control panel’s middle row contains the buttons for selecting the ranking metric. By default, the Occurrences metric is selected. Having a default metric provides a sorted initial state in the ranking panel. The Occurrences metric is the default because it seems to be the most intuitive metric to start with. The four ranking metrics buttons behave as a set of check boxes, meaning that the users can select just one or
Figure 2: Control Panel showing the basketball dataset selected, and the reference events for Chicago making or missing 3-point shots. The occurrence metric is selected, and the time bin size is set to 2 seconds.

Figure 3: The message shown when no dataset is selected yet.

Steps to start ranking patterns
The first thing you need to do is select a dataset from the Dataset dropdown in the Control Panel. After you do that, you can pick the Reference Event A and Reference Event B that you will be using to compare outcomes.

multiple of metrics simultaneously.

The intended work flow for EventScope to have users toggling these buttons, consequently altering the scores, and hopefully finding interesting patterns ranked at the top of the ranking panel. If none of these buttons is selected, the patterns have no score. In these situation patterns are alphabetically sorted.

The third row in the control panel shows the time bin size for the histograms (distributions). The initial value used for a given dataset is computed based on the length of the longest event sequence. The user can then modify this value in order to obtain smoother or more detailed distribution (i.e., bigger or smaller time bin sizes).

3.4.2 Ranking Panel
The Ranking Panel on the left contains multiple rows with charts for each event, the “discriminativeness” score, and the event name. In each chart two histograms are displayed, one for each reference event. Colors are kept consistent within the ranking panel and with the control panel: orange for the first event, blue for the second event. The vertical dotted red line represents the third (non-reference) event that the two histograms are aligned on.

Ordinate shows time of occurrences of the two events relative to the third event. The axis is split into time intervals, or bins, whose size can be adjusted from the control panel. If the data points represented simple occurrence counts per time bin along the abscissa, there would be dramatic differences in the histograms of events that occur more frequently from the ones that occur rarely, confounding the comparison. To avoid confounding, we use ratios of the reference event occurrences in that time interval to the total number of occurrences for that event type in the whole dataset.

Rows are sorted by the score. Each score’s background color varies from orange, to white, to blue. Bright orange shows that there is strong indication of relationship between the given row’s event with the first reference event, but weak indication of such relationship with the second one. Saturated blue indicates the opposite. Colors fade as the indicated relationship strength for the two reference events gets more similar or decreases.

At the top of the panel there are toggle-buttons controlling visibility of each reference event’s distributions. If one is switched off, the interface will hide the corresponding reference event’s distributions in all of the charts.

3.4.3 Details Panel
By clicking on any row in the ranking panel, the user selects that row’s chart to be magnified in the details panel, where it can be explored in more detail. The details Panel allows the user to zoom into any part of the chart by clicking it and dragging the mouse to define the zooming area. The “navigator” chart on the bottom gives the user an idea of
where the zoomed area is on the original chart. Two handles on the “navigator” can be used for controlling horizontal zoom. Hovering the cursor over the chart brings up values for both histograms at a particular time bin, along with the bin’s median time, in a tool-tip.

3.5 Implementation

EventScope was implemented as a web application, to assist in sharing the tool with others, for usability tests, asking for feedback, and even to allow real users to interact with it. Also, implementing EventScope as a web application provided the opportunity to use several open source graphical libraries copied from options.

AngularJS\(^1\) was used as the main framework for organizing the code and to bind the interface to the backend of the application. To create the charts, the Highcharts JS\(^2\) library was used. Highcharts JS was chosen for its simplicity, flexibility and support for all major browsers.

One of the challenges of building web applications is dealing with multiple resolutions depending on the size that users set for their browsers. A library called Bootstrap\(^3\) was used to build the layout. Bootstrap also helped to provide a flexible interface, so our EventScope displays nicely with different resolutions. Still, since its nature is to visualize several charts, EventScope requires a minimum display resolution of 840 × 600 pixels.

One of the design decisions EventScope made was to keep the control panel fixed when the users scrolled down searching for patterns. This way users can always see the details on demand, or filter out the data. Another overall design decision was to pick two colors to represent the two reference events. Those colors are consistently used through out the application, in the labels, buttons, charts and scores. Colors with high contrast were used so that users could easily identify what group data belonged to in a variety of conditions.

4. EXPERIMENTAL ANALYSIS

We tested the EventScope’s effectiveness by conducting usability tests and expert reviews. Through these reviews subjective user reactions and expert suggestions were gathered and leveraged to improve the tool. EventScope is being actively developed so a list of suggested changes remains critical to driving development in productive directions. It is relatively simple to find potential users with a knowledge of basketball rules for testing compared to other medical or business domain datasets. Using a dataset that is already understood reduces the amount of explanation that is required during the training phase of the usability study.

4.1 Dataset

Temporal datasets used in EventScope follow the EventFlow data format and consist of events that include a time stamp and a type of event. The events are part of sequences of events, also referred to as records, which are assumed independent of each other.

The dataset used in the usability test consists of events that occur in basketball games. Events were collected from a whole season of Chicago Bulls basketball games. Approximately 60,000 events were recorded for the entire dataset. 36 types of events were recorded for each basketball game, e.g., Chicago Defensive Rebound, Opponent Made Shot, and Opponent Jump Ball. The basketball events were organized so that each record consisted of either offense-to-defense positions or defense-to-offense positions, i.e. events starting from one team gaining control of the ball and ending with that team gaining the ball again. Breaking the data into records based on possession makes the assumption that events from different possessions do not have significant relationships with each other. Formatted, the basketball dataset is approximately 4.1 megabytes in size.

The types of events that are recorded for a given dataset can greatly affect the relationships that EventScope can extract. For example, the 3-point shot attempts were split into 2 events: “3-point shot made” and “3-point shot missed”. This data split enabled selection of made and missed 3-point shots as reference events for comparison. In some situations one is interested in shot choice and does not care about the shot’s outcome, and data may be preprocessed accordingly, which is out of the scope of this paper.

Data needs to be in JSON\(^4\) format to be loaded and read to be used with the EventScope tool. Currently, EventScope allows to select from a predefined set of datasets all available at the same server where it is hosted. Future versions of the program will allow a dataset to be loaded from any web-accessible location.

4.2 Usability Test

We conducted the usability testing on EventScope with a set of novice data analyst using the Chicago Bulls basketball dataset. Each user in the usability test was walked through a three step process. The process consisted of an introduction and consent phase, a training period, and an evaluation period. After all users had completed the usability test results were summarized and conclusions were drawn regarding the effectiveness of the tool. In order to assist in capturing user responses a Google Survey was setup and is available at online.\(^5\)

4.2.1 Introduction and Consent

The introduction and consent phase consisted of asking a potential user if they were willing to participate in a visualization tool evaluation for a computer science class project at the University of Maryland, College Park. It was made clear that participation in this usability test was voluntary, should take no more than 20 minutes, and that they could choose to stop at any time.

4.2.2 Training

The usability test participants were provided a short introduction to the tool using a sample dataset that had a medical theme and 4 events. This dataset was randomly generated. Different group members were performing independent usability test, so talking points and main concepts were recorded

---

\(^1\)http://angularjs.org
\(^2\)http://highcharts.com
\(^3\)http://getbootstrap.com/2.3.2
\(^4\)http://www.json.org
\(^5\)http://goo.gl/4D7hPJ
in the usability test web form. The general concept of aligning and scoring data relative to a reference event and the buttons in the user interface were explained at this point.

We use the medical dataset as an example. This sample dataset contains many randomly generated medical records, and each record is composed by many events. For example, a simple record could be: a patient took some medication ("drug1") on certain date and left hospital several days later ("discharge"), thus there are two events in that record, i.e. "drug1" and "discharge". It also could be a combination of "drug1" and "death" or "drug2" and "discharge". Given a number records like this, we may be interested in what events could potentially caused one outcome (e.g. "death") but not the other ("discharge"). In this case, events "death" and "discharge" are called "reference events".

There are three major steps to analyze the data using EventScope:

- **Step 1: Load dataset.** The "Medical" dataset is automatically loaded when users open the application. Users can later load another dataset by choosing it from the dataset drop-down menu. Currently only two datasets are available: "Medical" and "Basketball".

- **Step 2: Select reference events.** Click the "Choose Reference Events" button to set two different events as the reference events. For the medical dataset, it is a reasonable case to investigate "discharge" vs. "death" or "heart attack" vs "stroke". The drop down menus use different colors (blue and orange) that represent information relevant to the two different reference events. These colors are consistent in all the charts. Note that there are no limitations on which two events to use, so long as they are distinct and both occur in the dataset. They don’t necessarily have to be mutually exclusive.

- **Step 3: Check the events listed in the ranking panel.** Each chart shows distributions of the two reference events relative to event. The events are ranked by "discriminativeness" score in descending or ascending order. Users can click one event in the ranking panel and a larger figure will be shown on the details panel. The x-axis shows time difference between the selected event with the reference events, discretized using bin size. The density of the charts can be changed by tweaking the time bin size and unit.

Patterns can be interpreted from the top ranked events in the ranking panel. For example, the system can identify "drug1" as a high scored event since a large portion of "death" and very few of "discharge" happened after it. This pattern may indicate that the event "drug1" will lead to the event "death" but not "discharge".

4.2.3 Evaluation

Following the training period, participants were asked to perform two tasks using the basketball dataset. The first task was to identify events leading to a made 3-point shot but not leading to a missed 3-point shot. Specifically, they were asked to interpret the chart of the two reference events ("Chicago Made Shot 3" and "Chicago Missed Shot 3") aligned by "Chicago Offensive Rebound". Once the first specific task had been completed, participants were asked several less structured questions. They were asked to explore the dataset freely and write down any interesting findings. Before beginning to work on the second task, users were primed with a tip suggesting that they, “Try choosing different reference events and sorting the events differently.” Users were given approximately 5 minutes to answer this open ended question. The motivation behind asking users the less structured question was to determine, in an informal way, if EventScope aids in hypothesis generation. User reactions, comments and tasks answers were again recorded using the usability test web form.

In order to answer those question it is expected that the users will need to perform a specific sequence of operations. First, the user will need to split the data into two groups based on some outcome by selecting the appropriate reference events (e.g., made 3-point shot event and missed 3-point shot). Following the reference event selection, users may choose to adjust the width of the histogram bins. An appropriate ranking heuristic is the occurrence metric, selected by default. The user may toggle several ranking heuristics before settling on one.

4.2.4 Post Evaluation Questionnaires

Following the usability evaluation phase, several specific questions were asked to determine overall satisfaction with the EventScope tool. These questions have been used in conjunction with the open discussion during the usability testing, to determine the most significant areas where the EventScope tool needs improvements.

The questions asked include:

1. How would you rate your overall experience with EventScope? (on a scale of 1-5, 1 being poor and 5 being excellent)
2. Was EventScope easy to use? (on a scale of 1-5, 1 being hard to use and 5 being easy to use)
3. Are the charts intuitive relationship indicators? (on a scale of 1-5, 1 being not intuitive at all to very intuitive)
4. Which part of EventScope do you find most difficult to use or understand?
5. How do you think EventScope can be improved?
6. Do you have any additional Comments?

4.2.5 Results

We invited four participants to take the usability test, three of them were male and and one was female. All participants had a limited data analysis backgrounds but were moderately knowledgeable about the rules of Basketball. The study participants were mostly University of Maryland students with at least a bachelors degree in either Computer Science, Engineering or Business. None of the users were familiar with a alignment and ranking approach to temporal event mining. The group of users selected for the usability test represented our target population for the EventScope tool: amateur
Besides the interface, both experts emphasized the importance of using the interface to tell a story about the data. For example, an NBA statistician may be interested in the team’s performance after timeout. One expert searched for a specific pattern with the setting of “Chicago made shot 3” and “Chicago missed shot 3” as reference events, aligned by the “Chicago offensive rebound” event. She found the interesting insight that if “Chicago” team attempts a 3 point shot quickly (e.g., within one pass) after an offensive rebound they probably with miss it, but if they do it with some delay (e.g., after two passes) they will probably make the shot. It is very interesting to note that although our expert was specifically searching for this combination of events, EventScope automatically ranked this pattern as one of top interesting ones. This suggests that a novice user who were moderately familiar with the basketball data could have found the pattern.

5. DISCUSSION AND CONCLUSION

The usability testing of EventScope was completed. Non-expert user reviews of the tool were generally positive and all users were able to successfully complete assigned tasks. Remarks critical of the user interface were focused on the interface lag, initially understanding what the interface was showing and understanding how the metrics were working.

EventScope enabled the subjects of the usability test to correctly determine the most discriminative relations with the Chicago Made 3-Point and Chicago Missed 3-Point events. These relations matched user’s intuition regarding common sequences of events in basketball games. Verbally, users stated that integrating event outcome information into the pattern mining and visualization allowed users determine what happened under alternative scenarios. We imagine that exploring alternative hypothesis is a common task for users mining temporal event sequences.

Interface lag is a problem with EventScope. The size of the basketball dataset is pushing the limits of what can be manipulated with JavaScript running in a web browser. It is generally recognized that an interface reaction of 1.0 seconds is the limit for a user to progress with a thought uninterrupted [8, 14]. Loading the basketball dataset into the interface took on the order of 20 seconds and adjusting histogram bin sizes in the interface took approximately 5 seconds. Alternative implementations in Java or Python would have speed up the interface reaction time. Pre-computing certain values was considered, but was not implemented because it would not support quick exploration in new datasets.

EventScope usability testing revealed a detailed explanation of the temporal relation histograms on the left side of the visualization was required. It was not immediately obvious what the graphs were showing. Initial challenges could have been caused because these small graphs don’t have axis or labels because axis and labels would take up space and reduce the number of graphs that could be shown. Once users overcome the tool learning curve, they may appreciate EventScope’s selected configuration. Additionally, none of the subjects in the usability testing were familiar with EventFinder or PairFinder. If usability testing was conducted with users who were familiar with some advanced temporal mining tools, the learning curve may have been reduced.

We began our evaluation by giving a brief introduction and short tutorial of the EventFlow and then experts were allowed to interact with the tool on their own. We encouraged the experts to “think-aloud” and recorded their thought process.

The experts’ comments on the interface are summarized by category as below:

- **Guiding Message** One expert suggested the addition of guiding messages in the interface in order to suggest where to click and how to explore the data.
- **Metric Button** Both experts were confused by the names of metric buttons used for event ranking. The experts suggested use of non-technical words and pop-up tool tips on the metric buttons. They also suggested that finding a good example of a pattern that scored highly according to the metric to show in a help menu.
- **Ranking Panel** One expert suggested the left event should enable sorting events alphabetically by event name. Additionally, a text search box for easily selecting events was identified as potentially useful.
- **Axis Name** The experts were confused by the information displayed on the Y-axis. They suggested an easy to understand name for the Y-axis.
Usability testing identified the lack of feedback about what the ranking heuristics were computing as an issue. A qualitative explanation of what the heuristic is doing during a training period was satisfactory for non-technical people. In contrast, data scientist in the usability test tend to want to know exactly what the computation was that determined the list ranking. In order to facilitate user understanding of the rankings interface features could be added. Some of the metrics could highlight points on the graph, for example PeakRatio could draw circles around peak points on the graph. With the related OccurrenceRatio metric, it is less clear how to highlight points of interest. Adding visualization ranking features could be used to increase confidence in the visualization tool. In future releases, metrics for evaluating the strength of temporal patterns may need to be revised and selected based on the feedback they can provide to be displayed in an interface. In the current version, based on experts feedback, we simplified the weighted combination of measures to just selecting each measures (i.e., $\lambda = 1$ in eq. (4)).

Measuring an interface’s ability at assisting users in identifying novel temporal relation patterns is challenging problem. In usability testing, users are asked to perform specific task. These task generally do not involve hypothesis generation. Measuring performance on a task is a good way to determine if features are useful, however it fails to measure the ultimate goal of the tool which is hypothesis generation. A better understanding of tool effectiveness may have been acquired through a more complete study were hypothesis were generated and evaluated similar to the process presented by Saraiya et al. [13].

6. FUTURE WORK
There are several areas where EventScope could be upgraded, including; extending to multiple events, improving scoring metrics, and applying the visualization tool to new datasets.

EventScope currently identifies, scores and ranks temporal relations with only two events, limiting the sophistication of patterns that could be identified. Extending EventScope to handle non-reference event sequences would greatly increase the tool’s utility. Consider extending EventScope to handle two-event sequences: this would allow to find how the relationship would allow to find the “discriminativeness” of the patterns. Furthermore, currently users have to select the reference events to find the interesting patterns. This scenario is well suited for some domains like medical event mining. However, if the users don’t have an initial guess about what is interesting in the dataset, they still have to manually search through different reference event. An extension EventScope that could suggest potential reference events will be useful in such situations.

Analyzing new datasets with EventScope may lead to identification of new patterns in the data and additional features to add to EventScope. Several medical datasets have been acquired. There needs to be work done to understand the events and translate those events into a format that can be used by EventScope. Medical datasets may have more sophisticated patterns.

Currently, EventScope does not allow the user to perform transformations on the data. Types of events can only be defined when the dataset is recorded or pre-processed. Allowing aggregation or splits of event categories based on event attributes supplied with the data may lead users to explore more hypotheses. For the basketball dataset, aggregation would enable users to aggregate both the missed 3-point, 2-point, and 1-point shots into the same event type. Splitting events would enable users to separate the opponent block event into an LA Lakers block event, a Washington Wizards block event, etc.

Credits
All team members were highly engaged in idea inceptions for EventScope, weekly discussions and meetings, and were fully informed of all parts of the project. Below are specific credits in alphabetical order:

Andres Arellano - Helped in the design of EventScope. Defined the infrastructure for the software and implemented the main layout for the application. He took care of deploying the web app and wrote part of the paper.

Shobeir Fakhraei - Proposed the initial ideas for the project goal and scope definition. Met with experts to identify potential features. Organized acquisition and analysis of the data, and helped with ideas for interface design and ranking measures. Assisted in writing the initial proposal and the final paper. Wrote the video script and made the first version of the video.

Gregory Kramida - Designed and implemented many features in EventScope. Programmed dataset pre-processing and histogram generation. In the GUI, he formatted and properly arranged the charts. Wrote and edited several sections of the final Paper.

Hao Li - Designed and developed the usability test. Wrote the section of usability test and expert review. Recruited and walked through usability testing. Assisted in the development of histogram generation.
Michael Sorensen - Performed initial literature review. Assisted in the EventScope visualization development. Implemented several back-end services to compute the patterns’ scores. Recruited and walked users through usability testing. Wrote and reviewed several sections of the paper.

Acknowledgment

We would like to thank Prof. Ben Shneiderman for his guidance and support, and Prof. Catherine Plaisant and Megan Monroe for their mentor-ship and assistance during the development of EventScope. The datasets they provided were pivotal in both design and development, and their reviews allowed us to greatly improve the final product.

References


