ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity

Katerina Vrotsou, Jimmy Johansson and Matthew Cooper

Katerina Vrotsou, Jimmy Johansson and Matthew Cooper, ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity, 2009, IEEE Transactions on Visualization and Computer Graphics, (15), 6, 945-952.
http://dx.doi.org/10.1109/TVCG.2009.117

ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity

Katerina Vrotsou[*], Jimmy Johansson[*], and Matthew Cooper[*]

Abstract

The identification of significant sequences in large and complex event-based temporal data is a challenging problem with applications in many areas of today's information intensive society. Pure visual representations can be used for the analysis, but are constrained to small data sets. Algorithmic search mechanisms used for larger data sets become expensive as the data size increases and typically focus on frequency of occurrence to reduce the computational complexity, often overlooking important infrequent sequences and outliers. In this paper we introduce an interactive visual data mining approach based on an adaptation of techniques developed for web searching, combined with an intuitive visual interface, to facilitate user-centred exploration of the data and identification of sequences significant to that user. The search algorithm used in the exploration executes in negligible time, even for large data, and so no pre-processing of the selected data is required, making this a completely interactive experience for the user. Our particular application area is social science diary data but the technique is applicable across many other disciplines.

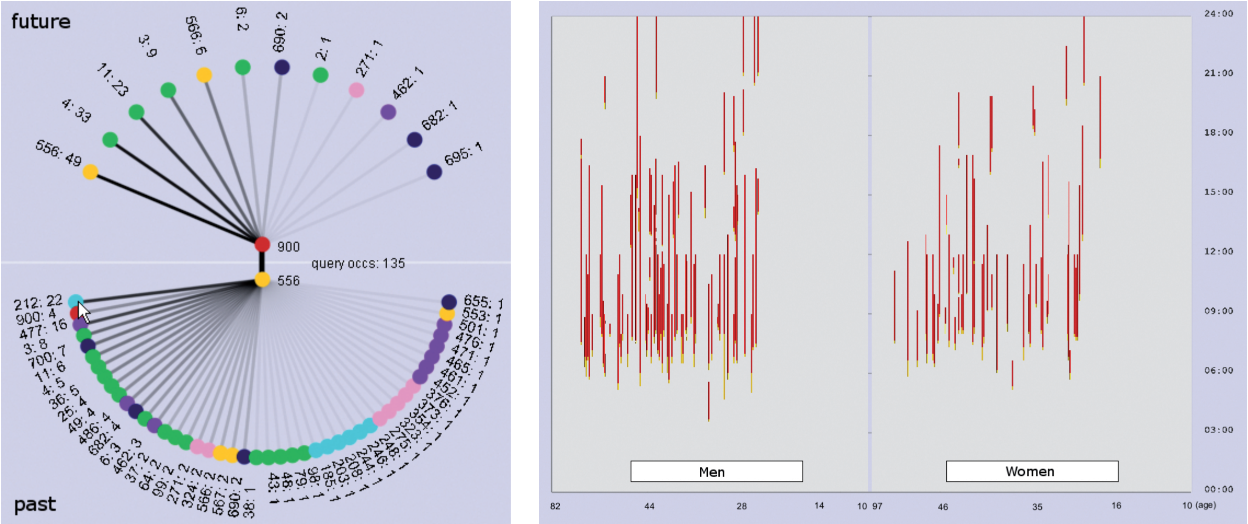
Keywords: Interactive visual exploration, event-based data, sequence identification, graph similarity, node similarity.

1 Introduction

Effective analysis of large and complex event-based temporal data is an area of rapidly increasing importance in today's information intensive society. The emphasis on development of methods for identification of sequences in such data makes evident the need for sophisticated analysis tools with uses in a wide range of fields. The most well-known use of sequence identification is perhaps in the analysis of sales information, where customers' purchases are tracked to identify frequently occurring sequences of purchases and so create models which can be used to predict potential future customer behaviour based on their previous shopping history. Other uses have been in the exploration of medical records to identify sequences and relationships correlating with good and bad outcomes, and in the exploration of web logs seeking patterns of accesses on sites to optimize the way in which information is presented. The focus of our own research in sequence identification is in the area of the social sciences where the establishment of patterns of movement and activities amongst large populations has found application in time use studies, social planning, urban design, occupational therapy and nursing care among others.

In the analysis of social science diary data, sequence identification has been attempted through a number of approaches. Pure visual methods are used, but these are mostly useful for small populations of subjects and therefore not sufficient in view of the rapidly increasing data sizes. Automatic sequence identification algorithms are also used, but such algorithmic approaches have a number of associated problems. The sequences are being sought because they exhibit a quality of 'interestingness' which makes them significant to the user doing the search. The nature of this quality is very dependent on the interests of that user, making it hard to encode in the search algorithm. As a result the search algorithms typically need to perform exhaustive searches of the activity space, which are very costly in terms of both memory and time, and then follow this up with human-centred filtering operations where the user explores the (usually extremely long) lists of sequences which have been identified. In order to avoid this problem, reducing both the computational complexity and the number

[*] Katerina Vrotsou, Jimmy Johansson, and Matthew Cooper are with Linköping University, Sweden, E-mail: firstname.lastname@liu.se.



(a) The ‘ActiviTree’ visual interface.

(b) Linked view showing a sequence in the context of the individuals’ daily life.

Figure 1: The two linked views which are the core of the application: (a) ‘ActiviTree’ visual interface. In the middle the currently explored query sequence (556 → 900: *travel by car → work*) is drawn. The activities connecting into the query sequence are drawn as connecting nodes in the bottom of the interface, while the activities connecting out of the query are drawn as nodes in the top of the interface. All nodes are ordered by significance score from left to right. Each node has a label showing the activity code describing it and its frequency of occurrence. Frequency of occurrence is also mapped to the opacity of the edges. (b) Linked view showing the currently explored query sequence (556 → 900: *travel by car → work*) in the context of the individuals’ daily lives. In this representation the individuals’ are drawn, ordered by sex and age, on the x-axis while the time is represented on the y-axis going upward. The representation reveals the daily activity sequences of each individual including the distribution, time of occurrence, and duration of the query sequence across the population.

of sequences found, simplistic measures such as requiring frequent occurrence of a sequence are applied. This is very effective for some kinds of problems, such as transaction analysis, where frequency of occurrence is the principal focus of the search. In the case of social sciences, however, as well as many other application areas, frequent occurrence is not the single major interest and, in fact, may obscure the object of the search: a small number of individuals whose behaviour is different from that of the bulk of the population. For these application areas frequency is not a good measure upon which to base restriction of the calculation and so the full analysis is required. In the case of our diary data sets this may require up to an hour of pre-calculation to exhaustively identify all sequences present in the data. Additionally most of these sequences are of no importance, representing trivial or obvious information, and the human filtering required to explore these sequences and eliminate the trivial is substantial.

In this paper we propose a tree-inspired interactive exploration environment for the systematic identification of sequences in social science activity diary data. We call our exploration tool ‘ActiviTree’. The method we introduce removes both the requirement to pre-process data as well as the need for human post-filtering. This makes it a powerful interactive tool which balances human interaction and data processing to yield efficient exploration of event-based data. We base our work on a matrix similarity algorithm, invented originally for the purpose of web searching, and adapt this to assign similarity scores to the events in our data which depend both on connectivity and frequency and are flexible to other weighting factors as well. These weighted significance scores, tuned by the user, are designed to reflect the importance of each sequence in their analysis.

The direction of the exploration depends on the user and sequences are identified stepwise, being increased one element at a time. The most significant single activities are first identified and ordered by their significance. The user can then select any activity as a starting point and the significance of all possible preceding and subsequent activities is calculated in a step requiring only a fraction of a second. These activities are then arranged as in and out branches, ordered by significance, to the selected activity (figure 1(a)). The user is then allowed to choose a particular branch, creating a linked pair, a pattern sequence. The same search algorithm is then used to identify the most significant activities preceding and subsequent to this pattern, enabling the user to create a sequence of three activities. This process is repeated, with the user adding a preceding or subsequent activity at each stage while simultaneously displaying the sequence in a separate linked view in the context of the entire data set as they work. The user can also remove activities from the current sequence, either at its head or tail, to move in alternative directions in the sequence-space. All calculations of significant connections are carried out during the interaction, with no apparent delay at each selection, making the process entirely interactive and enabling free exploration of the sequences present.

The main contributions of this work are:

- a visual data mining approach for the identification of significant event sequences
- an interactive exploration interface with great flexibility of use
- elimination of pre-processing and post-filtering in the mining process
- balanced integration of human interaction and data processing.

2 Related work

The visual data mining approach we have developed for exploring sequences in event-based data incorporates work from two distinct areas: visualization of event sequences and algorithmic sequence mining. Consequently we will consider both these areas separately to examine the related work and compare and contrast it with our methods.

Event-based data implies data in the form of events that have a start time and a duration, and occur sequentially. This differs from the area of time-series analysis, where continuous data changing over time is the focus, hence we do not include work from this area. A pattern in event-based data is defined as a sequence of events with certain attributes which make it interesting. Interestingness is not constrained to frequency, on the contrary a pattern that occurs infrequently may be much more interesting to identify since it is often also unexpected.

2.1 Visualization of event sequences

There are several examples of identification of event patterns using visual representations.

Torsten Hägerstrand [11] formulated a conceptual framework called time geography for describing human behaviour in time and space. In time geography an individual's movement is represented by a continuous vertical trajectory, the 'space-time path', within a three dimensional structure, the 'space-time cube', representing space in the horizontal plane and time in the vertical dimension. Several individuals' trajectories can be represented in the same cube and 'bundles' of trajectories can be identified indicating meetings of individuals and revealing patterns of spatio-temporal movement. Kraak in [14] implemented the space-time cube in an interactive geovisualization environment. Kwan has made substantial use of time geographical representations for exploring human activity patterns [15]. The space-time cube has also been used to represent and analyse other types of spatio-temporal data, for example paths of clouds [21] and discrete earthquake events [9]. Even though time geography is an intuitive and effective way of conceptualizing continuous change in time and identifying patterns, it quickly gets cluttered with increasing data size.

The time geographical framework was extended in [6, 7] to consider performed activities in time and the original geographical space was replaced with an abstract activity space. In this approach a continuous trajectory, the 'activity-time path', is used to represent the activities performed by an individual during a period of time. The activity-time paths are drawn within a simplified time-cube representation. Patterns of activities are, in this case, defined as sequences of activity events and are identified within such a representation by being highlighted in the activity space. [7] The work in [27] also focuses on representing and identifying patterns of performed activities using several representations. A 3D rod representation over a geographical map and 2D and 3D activity ringmaps are used, to show activity trends during the day. Another early example of finding patterns of events, in this case personal histories, using visual inspection is Lifelines, presented in [18]. In this work several aspects of a single history are listed as vertical time lines and distinct events as icons. Using filtering, highlighting and interaction tools a user can visually identify similar patterns between the time lines.

The approaches described this far for identifying patterns of events are purely visual and therefore sensitive to data size and subject to human error. Therefore visualization is often combined with automatic or semi-automatic techniques that can aid visual inspection. For example an extension of Lifelines was presented in [24] where multiple medical event histories can be visually explored and similar patterns can be revealed by aligning the data on discrete events. In [8] an interface is presented for finding temporal patterns across multiple histories by allowing users to construct complex queries and search the data for matches. A disadvantage of this method is that the creation of the queries is complicated, also the user must have some knowledge about the data in order to know what to look for, hence this framework is not well suited for identification of unexpected patterns. In [23] an apriori based algorithm was implemented for mining frequent sequences of activities in activity diary data. The results are visualized in a time geographical representation for the user to explore. The disadvantage with this method is that if the search is constrained then it overlooks infrequent sequences, while if no constraints are set then it identifies large numbers of patterns, many of which are trivial and uninteresting.

Event data can be visualized using directed graph representations, where each event constitutes a node in the graph and each transition from one event to the next constitutes a directed edge. Patterns in such representations are identified by considering the structure of the graph. Examples of the identification of patterns within graphs can be found in state transition and social network analysis. The nature of state transition data is closest to the activity data considered in this paper. In state transition systems it is interesting to locate sequences of states, similar to our interest in sequences of activities. Examples of methods for visualizing and exploring state transition systems using graph representations are found in [22] and [19]. The work in [20] is the most closely related representation to our work that we have found. The state transition system is successively clustered based on node and edge attributes. The clustered nodes are then represented twice in the visualization: once as source and once as target clusters, and labelled edges are positioned in between. A user can select a source, target or edge and explore their connection patterns. The visual representation used in [20] is rather complicated and even though it is certainly effective for the expert user, it can be complicated to perform a free exploration with no predetermined goal and the display tends to get very crowded due to the complex nature of such a system. In contrast, the approach described in this paper uses a tree-like representation to explore paths of nodes one at a time, leaving the display clear from clutter, and a separate display to show how the explored paths are distributed in the context of the total dataset.

2.2 Algorithmic sequence mining

The automatic discovery of interesting sequences in large sequential data sets has been an area of substantial research and there are many different mining approaches to this general problem depending on the type of data under study. The use of visualization in combination with such approaches is not as extensive, although several examples exist. We will, in this section, briefly present the methods most relevant to our research within social science diary data, namely three different approaches to frequent pattern mining, and sequence alignment.

Frequent pattern mining has been extensively researched since it was first introduced in 1993 [2] for the discovery of patterns in large transaction databases, so called market basket analysis. In 1995 [3] the approach was extended to apply to sequential data. An excellent overview of the frequent pattern mining research status is available in [12], hence we refer the interested reader to this work for details and will just present the methodology behind the most prominent approaches for sequential data of which there are three: the apriori method [3], pattern growth [17], and a vertical data format approach [26]. All three were initially applied to frequent patterns in transaction databases and later extended to sequential and structural patterns.

The apriori approach [3] is based on the property that a sequence of k items cannot be frequent unless all of its sub-sequences are frequent. So candidate sequences of increasing length are generated, stepwise, and checked for a minimum frequency against the database. The problem with this approach is that, even though the search space is greatly reduced, there can still be a very large number of candidate patterns that have to be pruned during the database scan. Also this method requires a new database scan for each new order, k, of sequences which is extremely time consuming. An example of this approach in combination with visualization for identifying, exploring and filtering patterns in social science diary data can be found in [23].

The pattern growth approach [17] mines frequent sequences without using candidate generation. Instead each of the set of frequent single items is used as a prefix and the database is compressed into a tree having a branch for each of the prefixes. Projected databases are created for each of the single items and patterns are then mined recursively in each of these separate databases. This approach significantly reduces the number of database scans needed, at the cost of a high memory load due to the large number of projected databases needed.

The vertical data format approach for mining sequences was introduced in [26] and is a method based on the apriori property for candidate generation and testing. However, during the initial database scan the data is transformed into a vertical format, meaning that for each single item a tuple with information about the locations in which the item appears in the data is saved. The great advantage of this method is that the database need only be scanned once, since all the information needed to evaluate the frequency of any candidate sequence is then available. The drawback here is, similar to apriori, that a large number of candidate patterns have to be tested and also the method does not apply to the identification of outliers. A visualization application using this approach on web log data is described in [25].

A fourth approach that is popular for mining sequential patterns in social sciences is optimal matching [1] or sequence alignment, originally developed for protein or DNA sequences [5]. Sequences are pairwise aligned and their similarity is computed. The results can then be classified using clustering algorithms to reveal trends in the sequential event data. An example of a visualization application based on this approach can be found in [10]. This approach is effective but is conducted in a pre-processing step and can be time consuming depending on the number of compared sequences.

The main drawback of the described methods is that they primarily search for frequently occurring sequences in order to reduce the search space with the risk that potentially interesting infrequent patterns and outliers are disregarded in the process. Our goal is to allow the user to decide on the interestingness of a sequence depending on the focus of their analysis and provide the flexibility to find all types of patterns. Furthermore, these methods identify patterns in a pre-processing step and the results are then made available to the user for exploration and filtering. The danger is that a great deal of computation is performed to identify a huge number of results most of which are completely uninteresting to the user performing the analysis.

The approach that we propose puts the user in direct control of the sequence identification process, requiring no pre-processing and no post-filtering. The patterns found at each stage are visually displayed with supporting data both as the ActiviTree representation and, simultaneously, in the context of the subjects' lives. The data mining algorithm is used as a tool which provides support to the process by ranking the potential sequences at each stage of the exploration, but presents all available options, ignoring nothing, and so even infrequent patterns and outliers are available for the user's consideration.

3 Interactive Sequence Identification

In this section we will describe the sequence mining approach and the means by which we adapt it for the creation of our interactive sequence identification method. We break this down into three sections, first we give a short description of the data, then we focus on the matrix similarity approach this work is based on, thereafter we describe how we apply this method within the proposed interactive sequence mining tool and its use in our social science diary data.

3.1 Social Science Diary Data

This research is concerned with exploring sequences of event data in social sciences. The data we are interested in consist of activities performed during the day by individuals in a population. Time diaries kept by a group of individuals have been collected and coded into a data set of distinct activities. Each individual's day is described by a linear sequence of activities. Each activity has a unique code, a start and an end time. This data can be considered as a directed graph between nodes, the activities, via edges, the transitions between the activities. Using this metaphor the order of the performed activities is preserved even though the actual timings of each are lost. Considering the data as being such a graph allows the use of algorithms originally developed for searching the internet, such as pagerank [16] and hubs & authorities [13], which act on such graph data. A generalization of hubs & authorities is used as a starting point in this work.

3.2 Generalization of Hubs & Authorities

Kleinberg in [13] proposed a method for retrieving web pages relevant to a given query which is based on the link structure of the Web. To do this he introduces the concepts of *hubs* and *authorities* through a *mutually reinforcing relationship*:

"a good hub is a page that points to many good authorities and a good authority is a page that is pointed to by many good hubs"

Using this relationship within an iterative algorithm a *hub* and an *authority score* are assigned to each page and updated at each iteration of the algorithm. If a page points to many pages with large authority scores then it should get a high hub score and if a page is pointed to by many pages with large hub scores it should get a high authority score.

Given a posed search query a collection of hyperlinked pages that are relevant to this query is first identified using a standard method such as query string matching. The ordering of the pages is then determined using the mutually reinforcing relationship. The collection of hyperlinked pages related to the query can be represented by a directed graph $G = (V,E)$ where vertex set, V, is the set of all pages, i, and edge set, E, is the set of all directed edges, (i, j), corresponding to a link from page i to page j. If h_j and a_j are the hub and authority scores of vertex j then the operations updating these scores are:

$$\begin{aligned} h_j &\leftarrow \sum_{i:(j,i)\in E} a_i \\ a_j &\leftarrow \sum_{i:(i,j)\in E} h_i \end{aligned} \tag{1}$$

As the number of iterations increases the scores converge. The resulting scores can then be ordered and filtered to return the largest scoring pages from each as the most relevant hubs and authorities.

Blondel et al. in [4] generalize the hub & authority scores to being similarity scores between the structure graph

$$hub \rightarrow authority$$

and the graph, G, of hyperlinked pages. So, the authority score of vertex j in G can be seen as a similarity score between vertex j and vertex *"authority"* in the structure graph and the hub score of vertex j as a similarity score between vertex j and vertex *"hub"*. Given this generalization the same mutually reinforcing iterative algorithm can be extended to apply to arbitrary structure graphs.

Consider a graph $G = (V,E)$ and the structure graph

$$in \rightarrow centre \rightarrow out.$$

Three scores are now assigned to each vertex j in G, an *'in-score'* (x_j^{in}), a *'central-score'* (x_j^{c}) and an *'out-score'* (x_j^{out}), and the updating operations of these scores are:

$$\begin{aligned} x_j^{\text{in}} &\leftarrow \sum_{i:(j,i)\in E} x_i^{\text{c}} \\ x_j^{\text{c}} &\leftarrow \sum_{i:(i,j)\in E} x_i^{\text{in}} + \sum_{i:(j,i)\in E} x_i^{\text{out}} \\ x_j^{\text{out}} &\leftarrow \sum_{i:(i,j)\in E} x_i^{\text{c}} \end{aligned} \tag{2}$$

The adjacency matrix B of graph G can be retrieved by setting each entry (i,j) of B equal to the number of links from vertex i to j. The updating operations can then be written in matrix form:

$$\begin{bmatrix} x^{\text{in}} \\ x^{\text{c}} \\ x^{\text{out}} \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & B & 0 \\ B^T & 0 & B \\ 0 & B^T & 0 \end{bmatrix} \begin{bmatrix} x^{\text{in}} \\ x^{\text{c}} \\ x^{\text{out}} \end{bmatrix}_{k}, \ k = 0,1,... \tag{3}$$

This can be written in compact form:

$$x_{k+1} = Mx_k, \ k = 0,1,... \tag{4}$$

where

$$x_k = \begin{bmatrix} x^{\text{in}} \\ x^{\text{c}} \\ x^{\text{out}} \end{bmatrix}_{k}, \ M = \begin{bmatrix} 0 & B & 0 \\ B^T & 0 & B \\ 0 & B^T & 0 \end{bmatrix}$$

and k is the iteration. Since it is the relative strength of the scores that is interesting, the normalized score values, z, can be used:

$$z_0 = x_0 > 0, \ z_{k+1} = \frac{Mz_k}{||Mz_k||}, \ k = 0,1,... \tag{5}$$

Any positive vector x_0 could be used to initialize this process. In this work we use the unit vector. As the number of iterations increases, $k \to \infty$, the score vectors, $z_1, z_2, z_3, ...$, ideally converge. However, the authors of [4] point out that z rarely converges to one limit but rather oscillates between an even and an odd limit. The even limit, given by:

$$z_{\text{even}} = \lim_{k\to\infty} z_{2k} \tag{6}$$

is selected as the definition of the similarity scores. The highest scoring vertices then constitute the most significant in, central and out nodes in the graph.

Considering our activity data as a directed graph, $A = (V,E)$, with vertex set, V, being the activity set and edge set, E, the set of activity transitions, this described methodology can be applied in order to assign significance measures to activities and successively explore activity patterns in a population. This is described in the next section.

3.3 Interactive Sequence Identification

In this paper we incorporate the generalization of hubs & authorities [4], described in section 3.2, into a visual data mining interface for exploring activity data and identifying patterns of activities. A pattern in the activity diary data context is a sequence of activities $(1,2,...,n)$ that, combined, express some interesting behaviour.

To give an overview, our method works as follows. The data is scanned and the significant single activities (*1-sequences*) are identified, these constitute the patterns of order 1. Selecting a *1-sequence* as a starting point, significance scores for each activity connecting to this sequence are calculated. Connecting activities are then drawn in a tree-like representation (figure 1(a)) in which the currently explored starting pattern is drawn in the middle and activities connecting to it are drawn as nodes linking into or out of it. Visual cues are given for significance and frequency of each node. A user can steer the exploration by choosing a connecting node to add to the identified *1-sequence* creating a *2-sequence*. This starts a new iteration of the algorithm and the nodes connecting to the new sequence are drawn. Having the user steer the creation of the sequences means that no pre-processing is needed. Also the algorithm does not need to have previously identified all $n-1$ sequences in order to find the n sequences. Given any sequence it will immediately calculate the significance of the activities connecting to it. We will now describe the details of the process.

3.3.1 Algorithm

The algorithm for identifying significant *n-sequences* has three steps:

1. Creation of an adjacency matrix.

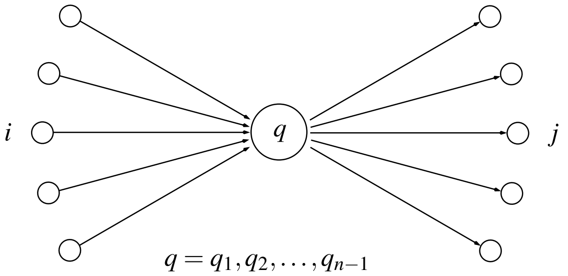


Figure 2: Example of links into and out of a query supernode q.

2. Repeated application of formula 5 to get similarity scores.
3. Weighting of the retrieved scores.

The initial step is to find all significant *1-sequences*. This prepares the ground for the exploration to start. Similarity scores are calculated between the nodes of the total activity graph, A, and the structure graph

$$in \rightarrow centre \rightarrow out$$

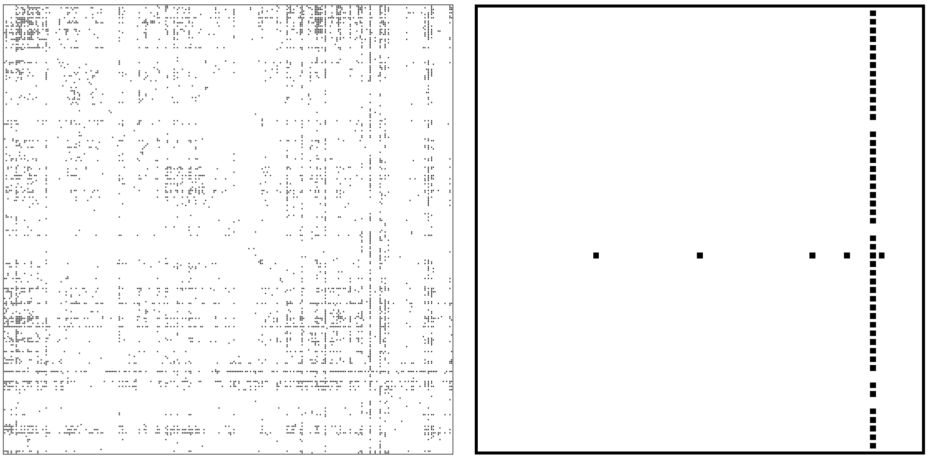
by first computing the adjacency matrix, B, of A. This is done by setting each element (i, j) of B equal to the number of transitions from activity i to activity j. The rows of B then represent in-links to, and columns out-links from, each activity. The mutual reinforcing iteration, formula 5, is then used to assign '*in-*', '*central-*' and '*out-scores*' to each activity node. We use a convergence criterion for deciding the appropriate number of iterations, the magnitude of k, in formula 5. For the data that we use this equation always converges within 10 iterations.

The similarity scores give an indication of the behaviour of each node within the graph, hence activities with a high *in-score* point to many significant activities, activities with a high *out-score* are pointed to by many significant activities and activities with a high *central-score* are both pointed to by, and point to significant activities. In the initial stage, a reasonable assumption to make is that the activity nodes with a high *central-score* are the most appropriate as patterns of single activities, *1-sequences*, since these are the most prominent 'link' activities holding the overall sequence of the individuals, in the data, together. Hence we choose the *central-score* as the significance score for the single activities.

The computed similarity scores are a measure of significance that is dependent on the connectivity of the nodes in the graph. Weighting this measure with the frequency of occurrence of the nodes gives a more general sense of significance in the context of the activity dataset. A node, for example, pointed to many times by a single low scoring node can be equally as interesting as a node pointed to once by each of several significant nodes. However, using only the similarity scores the former would not be considered a significant node. Hence, weighting the scores by frequency of occurrence balances these factors. Furthermore, since we are interested in the relative values of the scores we normalize them to the range $[0, 1]$. So now activity patterns of order 1, *1-sequences*, are found along with their significance scores and the exploration process of higher order patterns can begin.

In order to explore *n-sequences*, an *(n-1)-sequence* is used as a query sequence. A subgraph of the total activity graph, A, is found and the adjacency matrix, B, of this subgraph is computed. So to explore *2-sequences*, for example, a *1-sequence* is used as a query sequence.

The subgraph considered consists only of the currently explored query sequence itself, which can be seen as a supernode, q, and of activities linking into and out of the query sequence (figure 2). This is done by scanning the activity data set for matches of the query and considering the preceding and succeeding activities to these matches. An adjacency matrix, B, is computed by setting each element (i, q_1) of matrix B, where q_1 is the first activity of the query sequence, q, equal to the number of transitions from activity i to the query sequence and each element (q_{n-1}, j) of B, where q_{n-1} is the last activity of the query sequence, q, equal to the number of transitions from the query sequence to activity j. The rows of B then represent in-links to, and the columns out-links from, the query supernode. Figure 3(b) shows an example of such a subgraph adjacency matrix.



(a) Adjacency of total activity graph.

(b) Activities' subgraph adjacency.

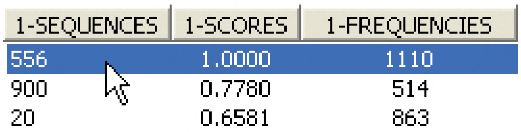
Figure 3: Example representations of adjacency matrices. (a) Adjacency matrix of the total graph of activities. Each element (i, j) in the matrix represents a transition from i to j. (b) Adjacency matrix of a subgraph of activities showing how these connect to a query sequence q. Each row element, (i, q_1), represents a transition from i to the first element in the query sequence, q_1, and each column element, (q_{n-1}, j), represents a transition from the last element in the query sequence, q_{n-1} to j.

Having computed the adjacency of the subgraph the mutual reinforcing iteration, through formula 5, is applied in the same manner as before and '*in-*', '*central-*' and '*out-scores*' are assigned to each activity node connecting to and from the query sequence. The objective now is, given an *(n-1)-sequence* query, to explore all possible *n-sequences*. So, we are interested in finding the activities linking into and out of the query and their respective scores, this means that we consider the activity nodes twice. Once as *in-nodes*, which are the activities having high *in-scores* linking into the query and once as *out-nodes* which are those with high *out-scores* linking out of the query.

The weighting of the scores also occurs in two steps. All *in-nodes* are weighted by the number of times each of them appears as an in-link into the subgraph, their frequency of 'in occurrence', while the *out-nodes* are weighted by the frequency of their 'out occurrence'. The weighted scores are then normalized to the range $[0, 1]$. As a result we get all nodes linking into the explored query (*(n-1)-sequence*) and their significance score (*in-score*) and all nodes linking out of the query and their significance score (*out-score*) and are now able to explore the combinations that can be made by adding these nodes to the query.

Alternative weighting by features of the data can be combined into the procedure in two ways, depending on the nature of the weighting to be applied. The adjacency values can be modified for each activity transition to reflect it's desirability. This is then combined into the calculation of the scores for each potential node. Alternative weightings can also be combined into the scores themselves at each iteration.

To summarize, the algorithm consists of three simple computational steps. An adjacency computation, a repeated matrix-vector multiplication and the application of weighting factors. The process of computing the adjacency matrix, B, expands linearly with the number of transitions between activities and so with the number of activities recorded in their diaries by the participants in the study. The dimension of the matrix, M, and the scores vector, x_k, in equation 5 are defined by the number of distinct nodes in the activity graph. In the case of our social science activity diary data this is a fixed value of approximately 330 different activity codes. Hence this step takes constant time and M occupies approximately 8MB. The matrix is very sparse so, in cases where the size of the matrix is much larger, sparse matrix methods could be used. The simplicity of these operations is what makes the exploration process highly interactive. Furthermore, any rejection of identified sequences is performed by the user. There is no cut-off by the algorithm, the identified sequences are simply ordered by their significance and made available to the user for filtering and analysis.



1-SEQUENCES	1-SCORES	1-FREQUENCIES
556	1.0000	1110
900	0.7780	514
20	0.6581	863

Figure 4: Part of the list of identified single activities.

3.3.2 ActiviTree

The sequence exploration algorithm is steered using an interactive visual interface in which the identified activity sequences performed by individuals in their daily life are represented in a tree-like fashion which we therefore call ‘ActiviTree’ (figure 1(a)). The ActiviTree is implemented as a feature within a visual exploration tool for activity diary data, details of which can be found in [7, 23]. In order to start exploring patterns in the data the significance of the single activities is initially computed and the results are presented to the user in an ordered list format, see figure 4. Clicking on an activity in the list will pop up the ActiviTree and the exploration process can start. In the ActiviTree time goes upwards.

The currently explored query activity sequence is drawn in the middle (as the trunk of a tree), the activities preceding it are drawn as nodes pointing to the query sequence (as roots connecting to the trunk) and the activities succeeding it are drawn as nodes pointed to by the query sequence (as branches growing from the trunk). Both *in* and *out* activity nodes are ordered by their significance score from left to right. So the leftmost activity is the most significant one. The frequency of their occurrence is mapped onto the opacity of the edge connecting to the query, so frequent sequences are opaque while infrequent ones appear more transparent. Each activity node has a label that includes its corresponding activity code and its frequency of occurrence as an *in/out-node* to/from the query. Right clicking on an activity node will show an explanation of the activity code. Nodes are colour coded in accordance with the activity classification colour scheme in [7].

Clicking on an in or out activity node in the interface will add it to the query sequence and identify the *in-* and *out-nodes* of the new query sequence in a fraction of a second. Clicking on the head or tail of the query sequence will remove the selected node from the query and update its *in-* and *out-nodes*.

If too many nodes connect to the explored query the representation can become cluttered and the nodes overlap. We avoid this by implementing a decluttering mode which shrinks the edge length of the nodes and expands a few at a time on mouse over (figure 5). Clicking on the nodes is available in this mode also, so the user is still able to select a node to add to the query and continue the exploration.

Each time a node is added to the currently explored query sequence the identified new activity sequences and their locations in the dataset are saved. Removing a node from the query sequence, and choosing a different exploration path will replace the saved sequences and their locations. The identified activity sequences can be highlighted in a separate linked view within the context of the individuals’ days in order to study their distribution more closely. This can be seen in figure 1(b) where the query sequence of figure 1(a) is highlighted. In the separate linked view time is represented on the y-axis going upwards and the individuals are represented ordered on the x-axis by sex and age. So, from left to right one can see older men to young boys and older women to young girls. This view reveals the distribution of the explored sequence across ages and sexes but also the duration of the sequence, its repetitiveness, and its time of occurrence during the day.

4 Usage Scenario

As an example of the operation of our application we describe the use being made of it in a study of social science diary data by our colleagues in the department of Social Sciences.

The exploration is initialized by computing the significant single activities. These are listed in the user interface (figure 4). The most significant activity is *travel by car* (556). We select this activity to start the ActiviTree representation. Since this activity occurs 1110 times in the interface, and is the most well connected activity in the data set, the representation is cluttered and the connecting nodes are impossible to distinguish.

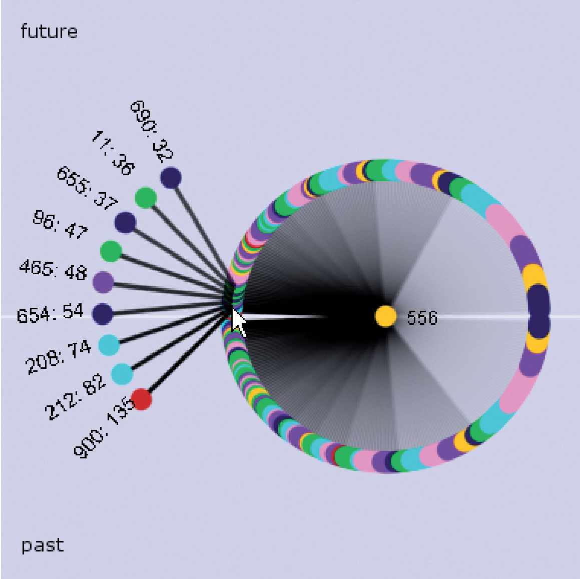


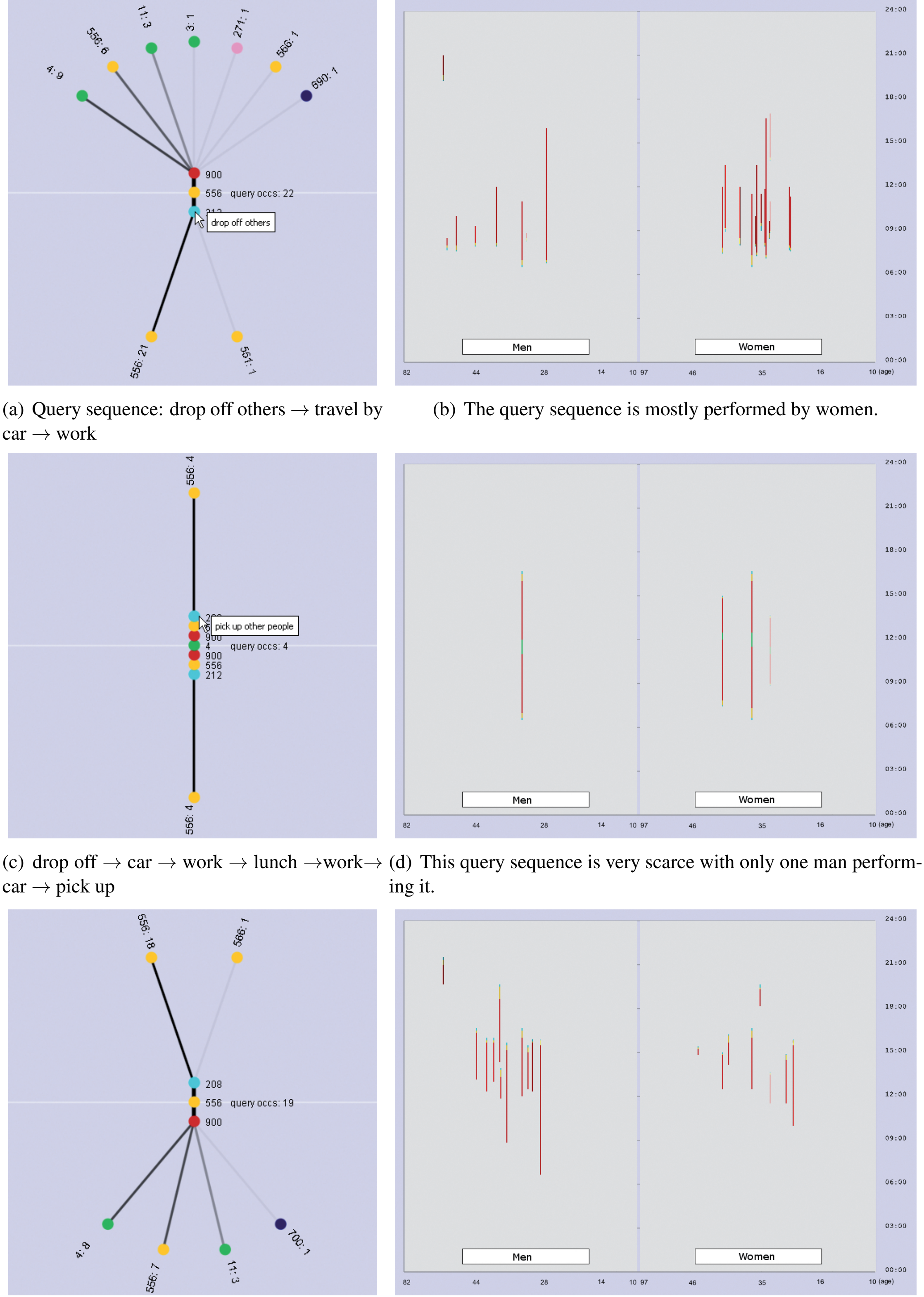
Figure 5: Example of all activities linking into and out of activity *travel by car* (556) in the decluttering mode.

We initiate, therefore, the decluttering mode (figure 5).

The most significant activity connecting out of *travel by car* is *work* (900). We select this activity and the ActiviTree updates (figure 1(a)). It is interesting, yet understandable, to see that many different activities precede the sequence *travel by car* → *work* while far fewer choices follow it. Obviously people do different things before getting into the car and driving to work while once they have started working the subsequent activities are mostly limited to travel by car, lunch and coffee breaks. Taking a look at the highlighted query sequence *travel by car* → *work* in the linked view (figure 1(b)) reveals its distribution. We notice that the spread of the sequence is quite even across the population. The pattern starts mostly in the mornings between 6:00 and 9:00 o'clock and is quite similar for men and women.

We continue by selecting the most significant activity connecting into the sequence *travel by car* → *work*, which is *drop off others*. in order to explore the behaviour of the sequence *drop off others* → *travel by car* → *work*. In the context of our activity data 'drop off others' usually implies dropping off children at school, day-care, friends, after school activities etc. Looking at the distribution of this sequence we see that the majority of individuals performing it are women. This constitutes an interesting behavioural pattern and raises the question of who then picks up the children. We follow this exploration path to find an answer to our question by adding the most significant nodes connecting out of the query. This leads us to the sequence *drop off others* → *travel by car* → *work* → *lunch* →*work* → *travel by car* → *pick up others* (figure 6(c)) which is performed by only four people of which one is a man (figure 6(d)). We realize that the query sequence we are exploring is very specific. Most people may not follow this exact activity pattern but rather also perform other activities in between. So, following our goal of finding out who picks up the children, we start making the query sequence more general by removing activities from its head until we notice an interesting distribution being revealed. When we have trimmed our query node down to *work* → *travel by car* → *pick up others* (figure 6(e)) we notice a change in the distribution (figure 6(f)): the new query sequence is performed mostly by men. This reveals a very interesting pattern of behaviour in the population sample under study. We have found that the distribution of labour concerning transportation of children in this population is quite even, but the women tend to take care of the morning 'shift' while men do the afternoon one.

This usage scenario is only a small piece of a much larger exploration process in which many interesting sequences were identified. In the selected example we started off with a single activity and worked our way in a chosen direction in the activity graph which resulted in the formulation of a question which we then explored by continuing the search until an answer was found. Other directions can be chosen during the exploration process that lead to other interesting observations, the forming of hypotheses and their verification or rejection.



(a) Query sequence: drop off others → travel by car → work

(b) The query sequence is mostly performed by women.

(c) drop off → car → work → lunch →work→ car → pick up

(d) This query sequence is very scarce with only one man performing it.

(e) Query sequence: work → travel by car → pick up others

(f) The query sequence is mostly performed by men.

Figure 6: Example of an exploration sequence. The left images show the ActiviTree. The right images show the distribution of the explored query sequence in the context of a group of individuals' days. The individuals are represented on the x-axis, and time on the y-axis going upwards.

5 Conclusions and future work

ActiviTree, combined with the powerful data mining algorithms behind it, provide a highly intuitive, interactive visualization tool for the exploration of event-based data and the identification of significant patterns of behaviour. Avoiding the need for time-consuming pre-processing, this tool allows the user to start immediately upon any problem and removes limitations on what they can find in any work session. The user can start in one area and find their search going in completely unexpected directions depending on the things they have previously identified. Removing the constraint of requiring the identified sequences to be frequently occurring also permits a different kind of searching from previous methods that have been available which, again, allows for a different approach to their work. The single individual who behaves differently from the norm is often the starting point for an interesting exploration session. The process of exploring this human-centred diary data is quite fascinating even for the lay-user and it is very easy to spend long periods exploring the data and finding many interesting features according to personal interests.

Initial testing with users has shown the tool to be preferable to previous work, which has been based on pre-processing approaches and the identification of frequent patterns, but it remains to be compared with the wide range of alternative approaches. This will be the focus of a study to be conducted later this year which will evaluate the ActiviTree approach as a method for pattern searching in diary data. The tool is also being developed for new search mechanisms, including more weighting factors and methods to identify patterns which are associated with energy consumption, a particular interest of one of our user groups. We are also looking into methods for saving the discovered patterns and the exploration process that led to them. This will make it possible for users to return to and continue along the same exploration path. Finally, we are interested in exploring other application areas including time and motion studies, and searches in medical records. These areas involve similar sizes of data but, often, much longer periods of time, greater variation in the durations of the events involved and much larger numbers of individuals. The system also has potential as a tool to explore social networks which can be much larger graphs but exhibit similar properties.

To conclude, we have presented an approach to the visual exploration of sequences in event-based data, in the form of activity diaries. The approach is based on an interactive visual interface for steering the identification of sequences, combined with an algorithm for computing the significance of the identified results. The greatest advantage of the presented idea is its simplicity which makes it both intuitive and interactive.

Acknowledgment

We wish to thank the Swedish Energy Agency for funding this work, Professor Kajsa Ellegård of Linköping University for helpful advice, and Statistics Sweden, for providing the data used in our studies.

References

[1] A. Abbott and A. Tsay. Sequence analysis and optimal matching methods in sociology: Review and prospect. *Sociological Methods Research*, 29(1):3–33, 2000.

[2] R. Agrawal, T. Imielisnki, and A. Swami. Mining association rules between sets of items in large databases. In *1993 ACM-SIGMOD International Conference on Management of Data (SIGMOD'93)*, pages 207–216, Washington, DC, 1993.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[4] V. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. V. Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, 46(4):647–666, 2004.

[5] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[6] K. Ellegård. A time-geographical approach to the study of everyday life of individuals - a challenge of complexity. *GeoJournal*, 48(3):167–175, July 1999.

[7] K. Ellegård and K. Vrotsou. Capturing patterns of everyday life - presentation of the visualization method VISUAL-TimePAcTS. In *IATUR - XXVIII Annual Conference*, Copenhagen, Denmark, August 2006.

[8] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, October 31 - November 2 2006.

[9] P. Gatalsky, N. Andrienko, and G. Andrienko. Interactive analysis of event data using space-time cube. In *Proceedings of the 8th International Conference on Information Visualization (IV'04)*, 2004.

[10] A. Godwin, R. Chang, R. Kosara, and W. Ribarsky. Visual data mining of unevenly-spaced event sequences. Interactive Poster in IEEE Symposium on Visual Analytics 2008, 2008.

[11] T. Hägerstrand. What about people in regional science? *Papers in Regional Science*, 24(1):6–21, December 1970.

[12] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:668–677, 1999.

[14] M.-J. Kraak. The space-time cube revisited from a geovisualization perspective. In *Proceedings of the 21st International Cartographic Conference*, pages 1988–1995, Durban, South Africa, August 2003.

[15] M.-P. Kwan. GIS methods in time-geographic research: Geocomputation and geovisualization of human activity patterns. *Geografiska Annaler: Series B, Human Geography*, 86(4):267–280, December 2004.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Stanford Digital Library Working Papers, Stanford, CA, 1998.

[17] J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M.-c. Hsu. Prefixspan mining sequential patterns efficiently by prefix projected pattern growth. In *17th International Conference on Data Engineering*, pages 215–226, 2001.

[18] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–ff. ACM, 1996.

[19] A. J. Pretorius and J. J. Van Wijk. Visual analysis of multivariate state transition graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):685–692, 2006.

[20] A. J. Pretorius and J. J. van Wijk. Visual inspection of multivariate graphs. *Computer Graphics Forum*, 27:967–974, 2008.

[21] U. Turdukulov, M. Kraak, and C. Blok. Designing a visual environment for exploration of time series of remote sensing data: In search for convective clouds. *Computers & Graphics*, 31(3):370–379, 2007.

[22] F. van Ham, H. van de Wetering, and J. van Wijk. Interactive visualization of state transition systems. *Visualization and Computer Graphics, IEEE Transactions on*, 8(4):319–329, Oct/Dec 2002.

[23] K. Vrotsou, K. Ellegård, and M. Cooper. Everyday life discoveries: Mining and visualizing activity patterns in social science diary data. In *Proceedings of the 11th International Conference on Information Visualization*, pages 130–138, Zürich, Switzerland, July 2007.

[24] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 457–466. ACM, 2008.

[25] A. H. Youssefi, D. J. Duke, M. J. Zaki, and E. P. Glinert. Toward visual web mining. In Visual Data Mining Workshop IEEE International Conference on Data Mining, 2003.

[26] M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[27] J. Zhao, P. Forer, and A. S. Harvey. Activities, ringmaps and geovisualization of large human movement fields. *Information Visualization*, 7(3-4):198–209, 2008.