Using action networks to detect change in repetitive patterns of action

Research-in-Progress

Brian T. Pentland
Michigan State University
East Lansing, MI 48824 USA
pentland@broad.msu.edu

Thorvald Hærem
Norwegian Business School
0484 Oslo, Norway
thorvald.harem@bi.no

Hamed Khaledi
Michigan State University
East Lansing, MI 48824 USA
Khaledi@broad.msu.edu

Abstract

We report on a simple method for detecting changes in repetitive patterns of action. The method involves computing the correlation of action networks before and after a focal date that is moved through an event log that represents the history of a process. Unlike process mining methods that focus on identifying an accurate model of a process at one point in time, we use a simplified model of the process to detect changes over time. The method can be used to objectively identify the date on which a process change occurred and also the relative magnitude of the change. We demonstrate the method with longitudinal data from two different organizations.

Keywords: Process change, action network, process mining

Introduction

In theory and in practice, organizations and processes are undergoing constant change (Tsoukas and Chia 2002; Günther et al. 2006; van der Aalst et al. 2012), and a great deal of this change is being facilitated by digitization (Bharadwaj et al. 2013). Research on the formation and dynamics of organizational processes and routines recognizes the need to understand the role of information systems (Pentland and Feldman, 2007; Pentland et al. 2012). At the same time, because processes are digitized, we have unprecedented opportunities to study them (van der Aalst et al. 2012). Taken together, these trends suggest that research on the impacts of digitization should have methodological tools for identifying process changes.

Here, we report the application of action networks as a tool for detecting and analyzing change in organizational processes. Action networks have been introduced as a way to model organizational routines (Pentland et al. 2010) because they provide a convenient way to represent the repetitive patterns of action that are characteristic of this phenomenon (Cohen 2007). While social networks are defined by ties between actors, action networks are defined by ties between actions. Research that uses action networks and the closely related concept of a “narrative network” (Pentland and Feldman 2007) has started to appear in the information systems literature (e.g., Goh et al. 2011; Yeow and Faraj 2011). The rapid growth of digitization of work processes makes it likely that more such research will appear in the future.

Action networks are a novel methodology in IS research. For action networks to become a viable tool in empirical research, we need to build a repertoire of well-defined concepts that be used to describe, analyze and interpret the structural properties of action networks. After defining the action network concept, we
introduce a method for the longitudinal analysis of process change and illustrate it using data from two organizations.

**Motivation: Detecting change in repetitive patterns of action**

Current process mining research recognizes that processes are constantly changing (van der Aalst et al, 2012), but current techniques emphasize the discovery of static models (van der Aalst and Weijters, 2004). With appropriate data, it is possible to recover an accurate, detailed model for a digitized process at any given point in time. However, longitudinal methods are not as well developed. Researchers interested in the dynamics of process change would benefit from techniques that enable the accurate detection of changes. The problem is particularly challenging because changes can be subtle and they occur in the context of data that is inherently noisy.

To address this problem, we need a representation of the phenomenon that allows comparison over time. We want to be able to represent a pattern of action, and variability within that pattern, but we do not necessarily need a complete event-state model of the kind used in current process mining (van der Aalst et al. 2012). Rather, we can use a representation that is simpler, such as an action network, which includes the minimum amount of sequential information about events in a process.

Figure 1 depicts a stream of events, indexed by time \((t)\). Each event is represented as an n-tuple of elements that are often used to describe events: who, what, when, where... This n-tuple has been applied in many disciplines, ranging from dramaturgical analysis (Burke 1969) to computer science (Fillmore 1975). We place an ellipsis at the end because in principle, additional facets can always be provided.

![Figure 1: Alternative network representations of the social world](image)

Each class of models in Figure 1 can be seen as a network, but with different labels on the nodes and arcs. Other classes of models can be constructed, as well. For example, established methodologies in process mining are used to construct Petri nets using data from process event logs (van der Aalst 1998). Because they are able to model concurrency and system state, Petri nets provide a more complete framework for process representation. The approach described here also relies on data from process event logs, but the action network representation is much simpler. Thus, while action networks would be inappropriate for business process conformance testing, they appear to provide a viable method for detecting process change.

**Action networks**

A simple way to project a stream of events onto a network is to focus on the actions or events. In such a network, the nodes represent categories of action, and the ties represent sequence. Like a social network, an action network represents relationship between pairs of actions. Like a simple linear narrative, an action network shows relations between actions, but it can be used to summarize a collection of narratives by aggregating pairs of actions. It is important to emphasize that action networks are not multimodal
(e.g., actors x action); they are unimodal (action x action). Because it only represents pairs of actions, an action network cannot represent concurrency.

Action refers to a step in a routine to accomplish an organizational task. Ties represent sequential relationships between actions. An action network can be summarized as a matrix, where the matrix elements, \( a_{ij} \), express strength of a tie from action \( a_i \) to \( a_j \). It can also be represented as a binary network, where \( a_{ij} \) is zero or one.

We are beginning to see networks of action used in empirical research in information systems. For example, Pentland et al. (2010) use the action network to empirically show that patterns of action generated by invoice processing are significantly different across four organizations which use the same technology for the routine. In research on healthcare information systems, Hayes et al. (2011) use action networks to study the impact of new technologies and potential needs for additional training at a medical center. Goh et al. (2012) use action networks to identify where and how health information technology influences patterns of work. Yeow and Faraj (2011) use action networks to investigated changes resulting from an ERP implementation. In all these cases, the network representation has been used to express the structure and variability in the pattern of action being studied.

These early studies suggest that this relatively simple representation can be useful when applied to patterns of action in the use of information systems. Since many readers are more familiar with social networks, it may be helpful to compare the structural properties of social networks and action networks. There are some important differences, as summarized in Table 1.

<table>
<thead>
<tr>
<th>Kinds of nodes</th>
<th>Social Networks</th>
<th>Action Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinds of nodes</td>
<td>Actors (e.g., individuals or organizations)</td>
<td>Categories of actions or activity</td>
</tr>
</tbody>
</table>
| Kinds of ties | ● No self-referential ties  
● Undirected or directed | ● Can be self-referential  
● Always directed (e.g., sequential) |

**Table 1. Comparison of social networks and action networks**

*Kinds of nodes*

In social networks, the nodes are social entities such as people or organizations (Wasserman and Faust 1994). Recent research has included some artifacts like information systems as nodes (Kane and Alavi 2008).

In action networks, the nodes are categories of action or activities (we use these terms interchangeable here). In terms of the stream of events in Figure 1, the nodes are the categories that would be used to describe “what” happened at each point in time. Unlike a social network, where individuals are treated as an irreducible natural kind, actions can always be subdivided into increasingly minute sub-actions (Abell 1987). Thus, when constructing an action network, there is always an issue of researcher judgment in defining the appropriate level of granularity. This is similar to the issue that arises in a traditional social network, where the researcher must define the scope (boundary) of the network (Wasserman and Faust 1994).

*Kinds of ties*

In social networks, different actants are connected by ties which can be summarized into two types: state tie and event tie (Borgatti and Halgin 2011). State-type tie represents social relations, which are continuous over time (Borgatti et al. 2009; Brass 2009). Event ties include the interaction or transaction between person A and B such as sending email or seeking advice. State ties can be measured in terms of strength, intensity, and duration. Event ties can be measured by frequency of occurrence (Borgatti and Halgin 2011).
In an action network, it makes more sense to conceptualize ties as being event-ties. Here, ties represent the sequential relation between actions, which can be measured by counting the number of times one action follows another. If these frequencies are normalized appropriately, they can become the conditional probability mentioned above.

In a social network, ties can be directed or undirected. In contrast, in an action network, ties are always directed, because they indicate sequence or causality. Furthermore, in a traditional social network, nodes generally cannot be connected to themselves. In an action network, however, self-referential ties are needed to represent the possibility that an action can repeat.

These differences in the nature of ties has implications for other basic network concepts, such as nodal degree and density. The degree of a node is defined as the number of ties this node has (Freeman 1979). When research in social networks uses undirected ties, there is no distinction between “in” and “out” degree. Because ties in an action network are always directed, in-degree and out-degree are always relevant.

Density describes the general level of linkage among the nodes in a network (Wasserman and Faust 1994). In social network, it represents the average strength of ties across all possible ties or what proportion of all possible dyadic connections are actually present. In undirected network, density is calculated relative to the number of unique pairs ($l/((n*n-1)/2)$; $l$ is the number of edges present). In directed network, density is calculated across the total number of pairs, so the density formula for a directed network is ($l/((n*n-1))$).

Since action networks are composed of directed ties, the appropriate formula should include all possible edges. However, since actions can repeat and nodes can be self-connected, the density formula needs to be based on all of the cells in the matrix: ($l/(n*n)$).

**Some concepts do not apply**

The dynamics of social networks are driven by mechanisms such as preferential attachment and transitivity (Snijders et al. 2010). However, these mechanisms are not readily applicable to action networks, because the nodes in an action network are not autonomous agents. When a social network grows, new ties often form based on preferential attachment (Barabasi and Albert 1999), whereby popular nodes gets more ties than unpopular nodes (so the “rich→get→richer”). As a result, a few key nodes will have many ties and a lot of other nodes will have a few ties. This process results in a network where nodal degree follows a power-law distribution. In contrast, in an action network, there is no reason to expect that nodes with high degree will necessarily form more ties.

**Using action networks to detect process change**

It is difficult to detect a change in a process or routine because even a stable process can produce thousands of unique sequences (Pentland et al. 2010). Furthermore, one has to expect some level of ongoing change (van der Aalst et al. 2010). Against this background, the challenge is to detect substantial changes.

In prior work (Pentland et al. 2011), we have used a simple split-half method based on the test for stationarity in Markov processes (Anderson and Goodman 1957). In this test, one simply divides the sample in half and tests the two networks for equivalence. While elegant, using a split-half test has some important limitations. First, it does not allow you to pinpoint when a change occurred. The split-half test can determine *if* a change occurred, but not *when* a change occurred. Second, it provides no information at all about the nature of the change, and very little useful information about the magnitude of the change. Thus, as a tool for understanding dynamics of organizational processes, the test for stationarity is not especially helpful.

The other conventional approach to identifying change is to interview participants. In process research, this has limitations. First, assuming that it is possible to identify and gain access to the appropriate participants, they will usually have a limited view of the overall process. They may understand their piece, but not the overall flow. This is why process mining research emphasizes the importance of making processes visible using objective, digitized data. Second, even if they have some awareness of the overall
Detecting Change in Repetitive Patterns of Action

It can be very difficult for participants to identify changes. It can be especially problematic to get an accurate retrospective account. Finally, while individuals may be aware of changes in policies or technologies, those changes do not necessarily correspond to changes in actual practice, which can be surprisingly resistant to managerial intervention.

Finally, as discussed above, current process mining tools can do an excellent job of extracting the current state of a process (assuming they have data of sufficient quality), but they are not as well suited for longitudinal comparison. There has been an effort to identify changes through automated examination of change logs (Günther et al. 2006). In principle, the use of objective data from the basic event log should be able to supplement these other tools for identifying and understanding process changes.

Methodology

As part of our on-going research on this topic, we have developed and tested a variety of different methods for detecting changes in repetitive patterns of action. The basic strategy has been to scan through the longitudinal data set and compare the process before and after some focal date, as discussed by Kifer et al. (2004). The challenge has been to devise a robust, meaningful comparison that can provide valid, reliable detection of "process change". Ideally, the technique should be fast, simple, and useful even with moderate quality data.

Within this general strategy, we have examined a wide variety of different metrics. We have considered raw frequencies versus rates of occurrences; different sizes of "window" before and after the focal date; different amounts of sequential information (e.g., 1-gram, 2-gram, 3-gram...), and different comparisons, including differences, absolute value of differences, sums of squares, ratios, and correlations. For a more complete description, interested readers can consult Khaledi and Pentland (2014). For brevity, we report here the most successful of these techniques, which is also the simplest. We refer to this as the Moving Bird, as shown in Figure 2.

Dataset and preliminary findings

To illustrate this method, we analyze data collected from the invoice processing system of two organizations. We will call them Site A and Site B. One site is a software company, and the other is an educational institution. They both use the same invoice processing software. The data were extracted from the event log of the workflow system that each organization uses to manage invoices. Site A had nearly six...
years of data available (9,570 invoices), while Site B had about two and half years (47,414 invoices). The sample is described in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Site A</th>
<th>Site B</th>
</tr>
</thead>
<tbody>
<tr>
<td>N sequences</td>
<td>9,570</td>
<td>47,414</td>
</tr>
<tr>
<td>Actions per sequence (Min/Avg/Max)</td>
<td>2 / 3.7 / 16</td>
<td>2 / 4.35 / 81</td>
</tr>
<tr>
<td>Days per sequence (Min/Avg/Max)</td>
<td>0 / 7.13 / 309 days</td>
<td>0 / 6.03 / 338 days</td>
</tr>
</tbody>
</table>

Table 2. Workflow event log sample

In both sites, there were a large number of invoices that were approved in one day or less with only two or three steps. At Site A, these very efficient sequences represent roughly half the data; at Site B, they represent about a third of the data. Also, in both sites, there were a handful of sequences with very long cycle times (several months). There were more of these slow sequences at Site B, which also had a quite a few sequences with large numbers of repetitive steps (mainly “reminder” steps). Readers may have already guessed that Site B was the academic institution.

The sequence of actions generated by processing each invoice provides the raw data for the analysis. It is like the stream of actions shown in figure 1, except that actions are “threaded” together for each invoice. The threading of one event to the next is what provides data about sequence and allows us to count the sequential ties between actions. Each action is described with additional data that we do not use in this analysis (who, what, when, plus additional fields that contain details that are not relevant for our analysis, such as the vendor and amount). Figure 2 shows a sample of 20 sequences from each site. Integers stand for actions, and we construct the action network by simply counting the pairs (e.g. 3-1, 1-7, 1-1, and so on). Since we are not comparing sites, the coding scheme is not important for this paper. In the interest of brevity, we do not report it here.

<table>
<thead>
<tr>
<th>Site A</th>
<th>Site B</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-1-1-7</td>
<td>6-2-9</td>
</tr>
<tr>
<td>3-1</td>
<td>6-2-9</td>
</tr>
<tr>
<td>3-1-8-1-7</td>
<td>6-10-2-2-9</td>
</tr>
<tr>
<td>3-1-7</td>
<td>6-2-9</td>
</tr>
<tr>
<td>3-1-8-1-7</td>
<td>11-2-9</td>
</tr>
<tr>
<td>3-1-1-7</td>
<td>11-2-9</td>
</tr>
<tr>
<td>3-1</td>
<td>4-2-2-9</td>
</tr>
<tr>
<td>3-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-8-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-1</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-8-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-1-7</td>
<td>4-2-9</td>
</tr>
<tr>
<td>3-8-1-8-8-1-7</td>
<td>4-2-9</td>
</tr>
</tbody>
</table>

Figure 2: Example sequence data
Detecting Change in Repetitive Patterns of Action

Figure 3 shows the result of applying the moving bird to the stream of sequence data at each site. The first thing to notice about the graphs in Figure 3 is the scale on vertical axis. At both sites, the majority of the correlations are above 0.9. From this, one can conclude that these networks (processes) are generally quite stable. However, on day 1112 at Site A, there is a spike, where the correlation drops to below 0.3. A similar spike occurs at day 517 at Site B, with a couple of smaller drops beforehand. Are these spikes meaningful? Do they accurately reflect a substantive change in the process?

<table>
<thead>
<tr>
<th>Site A</th>
<th>Site B</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Graph A]</td>
<td>![Graph B]</td>
</tr>
</tbody>
</table>

**Figure 3: Action network correlation over time**

Based on a careful inspection of the event logs from each site, the spikes in correlation do correspond to changes in the process. At Site A, the spike occurred in November, 2008. Leading up to that date, the action “POSTED” occurs rather sporadically. After that date, it occurs as the last item in every sequence. At Site B, the change occurred in November, 2010. As of that date, an action that had been occurring intermittently began to appear in nearly every sequence. Based on interviews with the personnel responsible for the workflow system, the change is not the result of a software upgrade. It reflects a change in practice. The smaller spikes leading up to the change in November appear to correspond to this intermittent introduction of the new work process. There do not appear to have been significant changes in cycle time, and this data does not contain any other outcome measures, so these process changes may or may not have had any practical consequences.

**Summary and on-going work**

The key contribution here is a simple methodology that allows us to objectively identify that a process change has occurred. This method appears to be quite robust with respect to parameter choices such as the size of the window and the amount of sequential information included. At least at these sites, using higher order sequential data does not alter the results. Further, because we are using correlation to detect the change, we also have information about the relative magnitude of these changes.

The technique reported here is part of our on-going work routine dynamics (Feldman and Pentland 2007), which includes the development of techniques for identifying and visualizing changes in repetitive patterns of action. Some directions for future research include the following.

**Characterizing process change.** We are also working on methods for identifying and describing the content of the change before and after process inflection points. Future work includes comparing the frequency of higher order sequential information (3-grams, 4-grams and 5-grams) before and after the focal date to characterize the nature of the change.

**Antecedents and consequences.** Now that we have a valid, reliable way to identify process changes, the next step is to identify their antecedents and consequences (Abbott 1992). The datasets from these
particular organizations do not include much useful information in that regard. The same is probably true of most digitized event logs, since they are usually intended to provide an audit trail for a particular process. Understanding antecedents and consequences will probably involve combining event logs with other sources of information.

**Visualizing other dimensions of process change.** Also, alternative methods of visualizing the streams of data seem likely to prove fruitful. The use of action networks pushes actors into the background, but it may be important to bring them back for some applications. Thus, the use of more sophisticated network structures (e.g., bipartite graphs that include actors and actions) may prove useful.

**Acknowledgements**

The research reported here was supported in part by the National Science Foundation (NSF SES-1026932).

**References**


