

Toward an open-source tool for pattern-based progressive analytics on interaction traces

Vincent Raveneau, Julien Blanchard, and Yannick Prié

Abstract—Due to the ever increasing use of data analysis tools, their performance has become an important scientific question, notably regarding the tools' ability to assist the analyst using them. We present our first proposals toward building a data analysis tool allowing an analyst to interact with his pattern mining algorithm, while working with interaction traces. First, we present an architecture for such a system, based on the ideas of *progressive analytics*. We then present our work toward a first implementation of this architecture.

Index Terms—Interaction traces, pattern mining, progressive analytics, user behavior

1 INTRODUCTION

In order to discover knowledge hidden in data, an analyst has to go through a succession of tasks known as the knowledge discovery process. Such tasks can be exploring the available data, formulating hypothesis and checking their correctness, or presenting parts of the data in a way that will highlight something interesting to the viewer.

If these tasks can be performed manually for very small datasets, the ever-increasing volume of data available has strengthened the need for tools able to assist an analyst in his work. Depending on research communities, different philosophies have been considered while tackling this problem. For example, the data mining community tends to focus on the efficiency of the algorithms processing the data, through measures such as memory consumption, complexity or execution speed. Although many papers acknowledge the difficulty and importance of considering the analyst running the algorithm, it is not an important part of the research effort. In the meantime, the visual analytics community has focused on ways to visually explore data, according to its focus on integrating the analyst in the knowledge discovery process [3, 7].

In this context, we seek to provide a tool mixing the visual analytics approach with pattern mining elements, giving the analyst the opportunity to interact with both the visualization and the pattern mining algorithm. Such tool would benefit from the visual analytics' consideration of the user, as well as from the efficiency of pattern mining algorithms. Other studies have already stressed the importance of merging pattern mining and visual analytics approaches [2, 10]. In order to provide the analyst with an interactive experience, we also want to integrate ideas from the progressive analytics approach to deal with the long waiting times pattern mining algorithms can induce.

We consider the case where an analyst is trying to discover user behaviors from interaction traces, i.e. sequences of events recorded during the use of a digital system by a user¹. By interacting with digital tools, everyone builds up a set of behaviors and practices. These behaviors are specific to each user, and continue to evolve with time. Having tools and techniques designed to facilitate the discovery of these behaviors could offer a better understanding of human-computer interaction and how the appropriation of digital tools work, as exposed

by Belin and al. [1].

In this paper we present our first proposals towards producing tools allowing an analyst to interact with both pattern mining algorithms and information visualizations in order to discover user behaviors in interaction traces, while complying with the ideas of progressive analytics. After presenting relevant existing work, we propose a global architecture for such tools. We then discuss our thoughts towards a first implementation of the architecture and the challenges that will need to be faced.

2 RELEVANT WORK

2.1 Using interactive pattern mining

Pattern mining is an exhaustive process, which produces understandable knowledge. Van Leeuwen [10] identifies four challenges when considering an interactive data mining process : taking into account the interestingness of a pattern for a specific user or task, instantly focusing the mining on subjectively interesting patterns, having principled evaluation methods for pattern mining algorithms and being able to visualize and interact with the discovered patterns.

While classifying existing scientific publications into three categories (mostly visualization with some algorithmic, mostly algorithmic with some visualization, evenly mixed visualization and algorithmic), Bertini and Lalanne [2] highlight the low number of works of the third category. Based on this observation, they express the need for more interaction between visualization and data mining, and provide some suggestions to do so.

2.2 Visualization and exploration of interaction traces

How to visualize interaction traces and how to explore them still remain open questions. In their work on this subject, Dou and al. [4] focused on interaction traces obtained while another analyst was working on some data. They have exposed that visually exploring interaction traces is an efficient way to reconstruct the reasoning process of the user whose actions are contained in the trace. In the process, they also find this technique to be a good way of recovering findings the original user had made. However, they identify that the potential lack of knowledge of the domain by the analyst can be a problem, and that not seeing the data the user was working on is problematic. This is especially true in cases where the user fixates on visually distinct data points, such as abnormal spikes, this information being hidden from the analyst.

Another work in this domain is that of Shrinivasan and van Wijk [8], in which they present a framework designed to build tools to explore interaction traces. Their proposal aims at supporting the reasoning process of an analyst working on interaction traces, by organizing the workflow around three elements :

- A data view, containing interactive information visualization tools
- A knowledge view, where the analyst can record analysis artifacts such as findings or hypothesis

- Vincent Raveneau is with University of Nantes, France. E-mail: vincent.raveneau@univ-nantes.fr.
- Julien Blanchard is with University of Nantes, France. E-mail: julien.blanchard@univ-nantes.fr.
- Yannick Prié is with University of Nantes, France. E-mail: yannick.prie@univ-nantes.fr.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

¹The recording task is handled by probes integrated into the system being used, and every event is associated with a time stamp. Depending on the system being used, the nature of the events and the information they contain may vary.

- A navigation view, where the analyst can link an artifact of the knowledge view to a state of the data view

With these tools, the analyst can go back to previous artifacts and their associated data view, to conduct his work on the data.

2.3 Progressive analysis of the data

Stolper and al. [9] proposed the notion of *progressive analytics*. This domain focuses on allowing an analyst to constantly be able to work on its data, an objective achieved through the use of algorithmic analysis of the data that regularly produces partial results. This always provides the analyst with some elements to work on without having to wait for the end of the entire computation. Based on the analysis of these partial results, the analyst must be able to steer the remaining analysis process without restarting from scratch. This approach is designed by opposition to the batch analysis, where the analyst must wait for the end of the process before seeing any results.

Due to their proposal being recent, few corresponding tools are available, even though Stolper and al. describe in their paper *Progressive Insights*, a system following their ideas. However, work is being done on this matter, such as Jean-Daniel Fekete’s *ProgressiVis* [5], a framework for the implementation of progressive analytics workflows.

3 AN ARCHITECTURE PROPOSAL

The underlying architecture we propose is illustrated in Fig. 1. From the interaction traces, a progressive pattern mining algorithm outputs patterns and statistical measures about them (such as their number of occurrences, occurrence durations or temporal distributions). Both the patterns and the measures are then stored inside a pattern manager, while the algorithm keeps iterating and producing partial results in the background. Patterns and statistical measures are represented in a visualization, alongside the events from the interaction traces. By exploring this visualization, an analyst is able to request additional patterns to continue his work. Such request is handled by the pattern search engine of the pattern manager, that will answer it in one or both of the following ways : by retrieving and sending already discovered patterns relevant to the query, or by steering the pattern mining algorithm’s execution in order to prioritize subparts of the data relevant to the query. Besides, some patterns are pushed in the visualization on the initiative of the pattern manager, thanks to a module which monitors the generated pattern stream. The module detects the unexpected or salient patterns on the basis of their statistical characteristics and of the user’s exploration behavior.

4 IMPLEMENTING THE ARCHITECTURE

Thinking about an implementation of the proposed architecture leads us to focus our attention on several aspects of the system. For each one of them, we present our vision of its implementation, and expose identified challenges.

4.1 The visualization

Due to the nature of the data considered, the visualization techniques used have to take into account the temporal dimension. This is primarily necessary for the representation of the interaction traces, but also for the representation of the occurrences of the discovered patterns. Different techniques can be used, depending on which is the most relevant for the information the analyst seeks to extract from the data. Visualizing the traces is generally done with a timeline-based representation. For patterns however, if a timeline can be used to locate occurrences, the associated statistical properties are better showcased with other representations (such as bar charts, curves...). Considering that patterns heavily rely on item names (or event types in our case), using visualizations such as trees or graphs can be useful.

Another important aspect that we think is important to consider is the added information brought by presenting different visual representation side by side, as it can help the analyst to establish connections between the different elements put in evidence by each view. This works even better if these representations communicate with each other, by propagating actions such as filtering or selection from one view to the others. This behavior is called *linking and brushing* [6].

For a first implementation of the tool, we consider three tasks for the visualization :

- Visualize occurrences of events or patterns through time
- Visualize every discovered pattern according to some constraints
- Visualize how patterns are distributed depending on their statistical properties

For each of these tasks, the analyst must be able to restrain the visualization according to constraints such as a period of time, a set of given patterns, a pattern template, a statistical property. . .

A challenging aspect of the visualization is to be able to update it with incoming patterns without disturbing the analyst exploring it. Updating the view in real time can lead to it changing too fast for the human eye. However, the analyst must be able to know when new data is available, and to request an update of the visualization. A possible way to seamlessly integrate new data is to do it when the analyst changes the view with operations such as zooming in and out, but the newly added elements would have to be distinguished from the previously existing ones. Creating this temporary distinction will allow the analyst to quickly see that new information is available for him to process.

4.2 The progressive pattern mining algorithm

To provide a pattern mining algorithm complying with the progressive analytics guidelines requires either to develop it from scratch, or to adapt an existing pattern mining algorithm to these guidelines.

In both case, two types of algorithms are possible, varying on their strategy to process the data. Breadth-first algorithms order the pattern extraction by length, starting with the shortest and gradually to the maximum size. Depth-first algorithms order the patterns according to

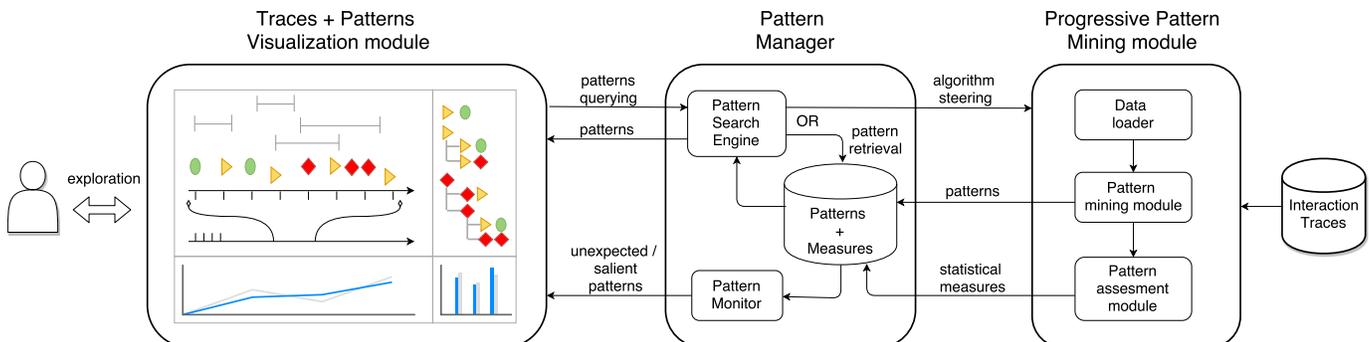


Fig. 1. Our proposed architecture

the vocabulary being used. This leads to all patterns sharing a common prefix being extracted before moving on to the next prefix. When considering the impact of these strategies on the analyst, the breadth-first approach starts by providing an overview of all the possible short-length patterns, which can be considered an overview of all the elements contained in the data. On the other hand, the depth-first approach starts by providing all the variations of a given element before moving on to the next one. Considering this, a breadth-first algorithm appears to be better suited for an exploratory search, while a depth-first one will be interesting for a study focused on some given elements.

Considering the steering of the algorithm, Stolper and al. [9] used a breadth-first algorithm to provide an overview of the dataset, while giving the analyst the opportunity to reorder the remaining exploration. In a first approach, we plan to do the same, before exploring other options in future versions of our tool. As being able to steer the pattern mining algorithm is really useful only if it can be done as early as possible, we believe that we should focus on providing the analyst with the information he needs to efficiently use this feature. This means providing an algorithm that will be quickly able to give an estimation of what could be the following patterns. To do this, techniques such as sampling over the data could be used, in order to provide an overview of the patterns.

4.3 Vocabulary of the patterns

The events contained in the interaction traces do not allow an accurate description of user behaviors, due to their low level. Pattern mining can be used to provide a higher level view of the data, considering that patterns can contain several events, but it does not involve the analyst in the process. A way to do so could be to allow him to specify similarities between different patterns, which could be used to enhance the suggestions made by the *pattern manager* after a request. For example, we can consider the case where events *A* and *C* have been declared similar by the analyst. If the analyst later shows an interest in patterns in which event *A* happens, the system could also suggest some patterns containing *C*. Considering that this could lead to unwanted results, it will be necessary to provide sufficient information to the analyst, allowing him to tell which patterns originate from his request on event *A* and which have been suggested by the system.

Another feature that can be useful to the analyst would be the ability to rename certain patterns. If this has no impact on the pattern mining, it helps with the interpretation of the results.

4.4 Suggesting patterns to the analyst

In order to assist the analyst in his work, we want the system to be able to suggest interesting patterns outside of the scope of the analyst's requests. The main difficulty with this task lies in the determination of what patterns the analyst deems interesting, especially without asking him first (which would then turn into another request rather than a spontaneous suggestion of the system). As exposed in Van Leeuwen's work [10], two types of interestingness can be distinguished: objective interestingness, computed from the data without any background knowledge or feedback from a domain expert, and subjective interestingness, derived from a specific user's goals and knowledge.

Dealing with objective interestingness is the easiest of the two, even if determining the appropriate quality measure to use can be tricky. To deal with subjective interestingness, we plan to follow van Leeuwen's recommendations, consisting of learning the user- and task-interestingness from the analyst's interaction with intermediate results. This could take place by analyzing which elements are ignored when steering the algorithm, or which items the analyst interacts with in the visualization. Other possible ways of determining subjective interestingness can be tied to specific features of the system, similarly to the similarity between events used as example in Sect. 4.3.

However, concept-drift, one of the major problems highlighted by van Leeuwen still stands as a challenge. If the analyst decides to change his center of interest in the data, the suggestion needs to be able to detect it and take it into account, in order to avoid considering previously important but now irrelevant elements. This is particularly

true in the case where the analyst has obtained satisfying information, and moves on to another task on the same data.

4.5 Accessibility of the system

Considering that very few tools affiliated with *progressive analytics* are available, we plan to release ours as open-source. Besides providing a working system, it will hopefully encourage further work on the subject among the community.

If we want our tool to be useful to analysts, providing an easy way to access it is important. As such, using a client-server model may be relevant. With the server's code being open-source, any analyst would be able to setup a server on a computer he controls, either to use it for himself or to make it available to others. Alternatively, providing a running server could allow those not willing to setup their own to remotely access the system to work on their data. As for the client side, making it accessible in a web browser would allow anyone to connect to a running server without needing to install any software.

If such a structure is to be used, it will be necessary to determine what part of the architecture will be located on the server or on the client. We suggest that the client handles the visualization part, while the server is in charge of running the pattern mining algorithm. This would notably allow the use of less powerful computers to work on the system, the mining cost being supported by the computer running the server. As for the *pattern manager*, giving it to the client could speed up the process of answering the analyst's requests, providing the relevant patterns are already discovered. However, a server-side *pattern manager* would present similar advantages to a server-side pattern mining algorithm, by reducing the processing power required from the client.

5 CURRENT STATE OF THE IMPLEMENTATION

Keeping these thoughts and challenges in mind, we implemented a first prototype of our architecture. In its current state, a server runs a progressive pattern mining algorithm looking for sequential patterns in interaction traces. The discovered patterns are then sent to a client to be displayed. This client, accessible with an internet browser, offers a timeline view containing the trace's events and patterns' occurrences. Other elements presented to the analyst are a list of all discovered patterns and statistical views on the data with interestingness measures such as lift. Interacting with these views allows the analyst to steer the mining algorithm while he works on the data.

6 FUTURE WORK

Now that we have a working prototype to experiment on, we plan to test its ability to handle large amounts of data. We will then iterate on it to enhance its capabilities, by adding features such as new visualizations and mining algorithms. Another element to consider for future work is the *pattern manager*'s ability to suggest relevant patterns to the analyst. To this end, exploring ways to determine the relevance of a pattern will be necessary.

REFERENCES

- [1] A. Belin, Y. Prié, and A. Tabard. Supporting the Development of Digital Skills. In *Digital Intelligence*. Nantes, France, Sept. 2014.
- [2] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.
- [3] K. A. Cook and J. J. Thomas. Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2005.
- [4] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning process from user interactions. *IEEE Computer Graphics & Applications*, 29(3):52–61, 2009.
- [5] J.-D. Fekete. ProgressiVis: a Toolkit for Steerable Progressive Analytics and Visualization. In *1st Workshop on Data Systems for Interactive Analysis*, p. 5. Chicago, United States, Oct. 2015.
- [6] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002. doi: 10.1109/2945.981847

- [7] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering the information age-solving problems with visual analytics*. Florian Mansmann, 2010.
- [8] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1237–1246. ACM, 2008.
- [9] C. Stolper, A. Perer, and D. Gotz. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, 2014. doi: 10.1109/TVCG.2014.2346574
- [10] M. van Leeuwen. Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pp. 169–182. Springer, 2014.