Process Mining for Clinical Workflows: Challenges and Current Limitations

Martin LANG\textsuperscript{ab1}, Thomas BÜRKLE\textsuperscript{a}, Susanne LAUMANN\textsuperscript{b}, Hans-Ulrich PROKOSCH\textsuperscript{a}
\textsuperscript{a}Chair of Medical Informatics, University Erlangen-Nuremberg, Germany
\textsuperscript{b}Dpt. of Image and Knowledge Management, Siemens AG Medical Solutions, Germany

Abstract. Process mining is an emerging technology in the context of Business Process Management with the goal to derive process models from observed system behavior. The global goals are: to detect previously unknown process structures, to implement consistent process controlling which may involve computation of realistic cycle times and frequency of occurrence of process pathways, or to quantify the conformance to guidelines. We did a detailed hands-on evaluation and analysis of established process-mining approaches and assessed their abilities to cope with the challenges of clinical environments. None of the examined 7 approaches fulfilled all requirements, but 2 could be circle out, which are to some degree suitable for clinical process mining.

Keywords. Process Mining, Clinical Workflows, Evaluation

Introduction

Due to increasing financial pressure and DRG based reimbursement, executives and controllers in a clinical environment are interested in optimized clinical and administrative processes. Often however, there is no knowledge about those processes and typical process analysis in a hospital environment is time consuming and tedious. Process variations are common and without rigid clinical pathways and a thorough deviation analysis their occurrence and frequency remains unknown. Shifts in workflows may go unnoticed for weeks or months. Process Mining has the potential to provide this kind of information by deriving process models from observed system behavior, like e.g. system log files. Currently, a broad variety of approaches are available [1-13], but the characteristics of clinical processes are challenging current solutions. In this paper we present a detailed analysis of existing mining approaches in respect to their ability to derive process models from real-life clinical data.

\textsuperscript{1} Corresponding Author: Martin Lang, Chair of Medical Informatics, Friedrich-Alexander-University Erlangen-Nuremberg, Krankenhausstraße 12, D-91054 Erlangen, Germany; E-mail: martin.lang@m-lang.de.
To assess a selection of current process mining solutions we used data extracted from the log files of the radiology information system (RIS) and data of the hospital information system of the university clinic Erlangen, Germany. We examined event data from all ~15,000 Computer Tomography (CT), Magnetic Resonance Imaging (MR), ultrasound (US), and X-ray (XR) examinations performed in 2006. In this study we used the ProM-Mining Framework v4 [14] with 7 up-to-date algorithms. We analyzed and assessed the \( \alpha \)-algorithm according to van der Aalst et al. [1], the \( \alpha++ \)-algorithm according to Wen et al. [2], the heuristic-mining algorithm according to Weijters et al. [3], the DWS-algorithm according to Greco et al. [4-5], the multiphase-algorithm according to van Dongen and van der Aalst et al. [6], the genetic-mining algorithm according to de Medeiros et al. [7], and the theory-of-regions-based algorithm according to van Dongen and Busi et al. [8], as provided in the ProM tool. The study focused on the ability of the different approaches to derive correct process models from real-life clinical data and their ability to cope with the challenges of clinical processes. We chose clinical imaging as a research environment due to its high penetration with IT applications - able to deliver detailed logging information - and because clinical imaging processes compare well with other clinical processes. We analyzed and assessed the following criteria:

- Can the method deal with incomplete and noisy input data?
- Is it able to distinguish sequences, forks, and concurrency?
- Is it able to derive block-structured and arbitrary loops?
- Is it able to distinguish repeated activities from activities in loops?
- Can it deal with fuzzy process entry and end points?
- Is it able to detect different process types and variants?

To answer those questions, data was extracted from the log files and the database of the RIS, and transformed into a representation suitable for the mining algorithms. To assess the ability of an algorithm to induce complete and correct models, the improved continuous semantics fitness measure (ICS) [7] and the source data verification (SDV) methodology were used. The ICS fitness quantifies the ability of an algorithm to induce a model \( M \) covering the behavior given in a log \( L \) (completeness). It describes how good a process model covers the given behaviour. The ICS fitness assumes values between \( (\alpha; 1) \) (1 is optimum) and is defined as follows with \( f_p = a + b \) [7]:

\[
ICS(L, M) = \frac{|Parsed Activities(L, M)| - f_p}{|Activities(L)|}
\]

\[
a = \frac{|Missing Tokens(L, M)|}{|Instances(L)| - |Instances With Missing Tokens(L, M)| + 1}
\]

\[
b = \frac{|Tokens Left Behind(L, M)|}{|Instances(L)| - |Instances With Tokens Left Behind(L, M)| + 1}
\]

To assess the preciseness of mined models the SDV methodology was used. Therefore subgroups as e.g. routine CT or emergency CT were selected in an iterative refinement process and the corresponding mined models were analyzed. We checked if the mining...
would deliver artificial pathways. We e.g. choose a process with repeated activities to assess the capability of the algorithms to correctly distinguish repeated activities from activities in loops. The same course of actions was carried out to evaluate the other criteria like being able to deal with fuzzy process entry and end points, etc. If the induced process models allowed for artificial behavior, the corresponding algorithms were devaluated. A comparison of mined models with a reference process was not realized because binding reference processes are mostly missing, deviations are common and the behavior given in the logs is not necessarily complete.

2. Results

The study resulted in over 120 process models derived from the source data with and without filtering, selecting subgroups and so on. The details will be published in [15]. Here we present 2 typical examples for mining regular CT examinations with the α- (Figure 1) and the DWS-Algorithm (Figure 2). From those 2 models, which have been derived from identical source data, it becomes immediately clear, that some algorithms perform better, whereas others are not even able to detect a linear workflow of more than two activities. The DWS algorithm in Fig 2 produced a fairly good model, showing e.g. that from “Create order”, most cases will go to “Document procedure” and to “Prefetch previous images”. In the second case the workflow will continue in parallel with “Store images to PACS” and with “Create report” to finish with “Sign off main report”. In some cases however, images will be deleted from PACS or the report will be edited. For such a model we would now check if all direct connections between any two activities did exist at least for one patient in the source data. We obtained the following results: Models constructed using the approaches of Wen et al., Weijters et al., and Greco et al. (see also fig 2) showed only elements, which are true to the reality. The genetic miner of de Medeiros et al. – when used with heuristic initialization - did also produce correct models. When used with random-based initialization however, it showed artificial behavior under certain conditions. Though artificial elements will be penalized during the stepwise genetic evolution of the models, they cannot be ruled out
completely, especially if only a few iterations are performed. We noticed artificial behavior also in the multiphase approach of van Dongen and van der Aalst, which e.g. would generate artificial process start and end points. The \( \alpha \)- (see Figure 1) and the -region-miner-algorithm did generate none or incomplete models. The evaluation resulted in the following weighted and averaged ICS values for the mined CT, US, MR and XR models. \(-1,249 \text{ for van der Aalst et al., } -1,305 \text{ for Wen et al., } 0,683 \text{ for Weijters et al. and Greco et al., } -0,733 \text{ for van Dongen et al., } 0,919 \text{ for de Medeiros et al., and } -1,294 \text{ for van der Aalst and Rubin et al. All algorithms except the } \alpha \text{ and the region-miner-algorithm were able to derive forks, joins, as well as sequential and parallel behavior from the input data. Here the problem is that the } \alpha \text{ and region-miner-algorithm require absolutely noise-free and complete input data, which cannot be assured when clinical data from log files is used, despite elaborate ETL (extraction, transformation and loading) methods used in this study. Four of seven algorithms (\( \alpha++ \), heuristic-, DWS- and genetic-algorithm) were able to correctly detect arbitrary and block-structured loops. One algorithm (van Dongen and van der Aalst et al.) derived loops, which did not correspond with the input data, whereas the \( \alpha \)- and region-miner again did not produce sensible mining results in this task. The differentiation between repeated activities and activities in loops is an issue for the majority of the algorithms. Only de Medeiros et al. propose a variant of its genetic-mining algorithm, which is (partly) able to handle these constructs, by using the sets of successors and predecessors to distinguish between repeated activities and activities in loops. This implies a limitation to activities and excludes repetitive sub-processes, because the predecessors and successors within these sub-processes are not distinguishable from each other. All other investigated approaches were not able to handle these constructs, neither repeated activities, nor repetitive sub-processes. The efficient handling of fuzzy process entry and endpoints is another challenge for the approaches. Again, the \( \alpha \)- and region-miner algorithm did generate insufficient results. In contrast, both heuristic- and DWS-miner perform well in this regard by counting successor frequencies globally. But this straightforward computation of relative successor frequencies (using activity

<table>
<thead>
<tr>
<th>Assessment criteria</th>
<th>van der Aalst et al. (( \alpha )-algorithm)</th>
<th>Wen et al. (( \alpha++ )-algorithm)</th>
<th>Weijters et al. (heuristic-miner)</th>
<th>Greco et al. (DWS-algorithm)</th>
<th>de Medeiros et al. (genetic-algorithm)</th>
<th>van Dongen et al. (Multiphase miner)</th>
<th>van der Aalst and Rubin et al. (Region miner)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth to reality in contents</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+/-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Noise and incompleteness</td>
<td>-</td>
<td>+/</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sequences, forks, and concurrency</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Loops</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Repetitive activities</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+/-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Fuzzy entry and end points</td>
<td>-</td>
<td>+/-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Process types and variants</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
labels as identifiers) sacrifices the ability to distinguish repeated activities from activities in loops. This also holds true for the multiphase-algorithm. Besides, the latter generated artificial start tasks. The genetic-algorithm uses a guided evolution of potential process models to find the best fitting one for a given dataset. It tends to generate models representing the most frequent behavior of the input data and handles fuzzy process entry and end points well. Detection of different process types and variants was only possible with the DWS-algorithm. It uses a preprocessing step to cluster event traces based on string distance functions. The idea is, that process instances of the same type or variant have similar event traces. Actually, this preprocessing step is independent from the algorithm itself, but firmly integrated only in the DWS-algorithm. The other 6 algorithms failed on this task. A summary of the study results is given in Table 1.

3. Discussion

Analysis and evaluation of 7 different process-mining approaches revealed, that most mining algorithms have problems when analyzing event data from clinical workflows. This results either in failure to construct any valid process model or in models, which do not reflect reality. We could obtain correct (but portionionally incomplete and fractional) results only from 4 algorithms, namely Wen et al. (α++), Weijters et al. (heuristic), Greco et al. (DWS), and de Medeiros et al. (genetic). One reason is that even clinical log data, which undergoes a thorough ETL process, is still to some degree noisy and incomplete, meaning that some data parts are missing or timestamps are incorrect because clock time might differ a fraction between modality and RIS. Thus algorithms, which operate with Boolean decisions to induce dependencies between activities, develop problems. One single negative data item in the log (e.g. caused by noise or incompleteness) will prevent correct detection of a dependency despite many hundred correct data items. In our study the α-algorithm, the multiphase-algorithm, and the theory-of-regions-based algorithm were badly affected by incomplete and noisy data. Surprisingly, the α++ algorithm, which is also known to require noise-free and complete information, performed fairly well on our data set. We suspect this is due to the Boolean isIndirectlyFollowedBy-Relation, which is less susceptible to mine wrong or none edges in the presence of noise and incompleteness compared to the isDirectlyFollowedBy-Relation used e.g. in the α-algorithm. The heuristic-miner algorithm, the DWS-algorithm, and the genetic-miner algorithm produced good results despite noisy data. The challenge to correctly induce process models from clinical event data aggravates, if activities occur multiple times in the process models, without being part of a loop. In combination with noisy data, only the approach of de Medeiros et al. was capable of coping with this fact, but at the expenses of considerably higher computation times in the range of multiple hours for the data sets of this research. Only the approach of Greco et al. was able to detect process types and variants. This study does not cover all published process-mining approaches. The approaches from Cook and Wolf [9], Agrawal et al. [10], Pinter et al. [11], and Wen, Wang and Aalst et al. [12] were not considered, because of missing implementations or excessive information demands (e.g. requires explicit information about concurrency). Nevertheless, we think that the 7 algorithms tested in this study are a representative selection of available approaches, comprising the latest and most common approaches of the process mining research community. We conclude that none of the discussed approaches is able to
meet all major challenges of mining clinical processes such as noise, incompleteness, multiple occurrences of activities, or richness of process types and variants. The approaches of Weijters et al. or Greco et al. produced good results in the presence of noise and incompleteness combined with acceptable computation times in the range of minutes. Extending these approaches to enable them to cope with multiple occurrences of activities will be subject of future work. Apart from the fact that we considered only data taken from one hospital, almost every clinical information system provides such kind of logging data and therefore process mining can be applied. Even if ETL efforts, data granularity and quality may differ, the underlying characteristics of clinical workflows remain the same, regardless of the concrete infrastructure. Despite the mentioned flaws of current approaches, the concept of process mining carries great potential in helping to understand everyday clinical workflows and their variations. This will be helpful not only for process monitoring and controlling, but may eventually give us a better retrospective understanding of relations between diseases and corresponding treatment processes, once we are able to derive (partial) treatment process models merely by pushing a button.

References