An Improved Process Discovery Approach Based on the Markov Transition Matrix

Hong Li¹,a,* and Hao GAO²,b

¹,²Southwest Forestry University, Yunnan, China
ᵃlh_1985@163.com,ᵇ1034240978@qq.com

Keywords: Process mining, Markov transition matrix, Process reconstruction.

Abstract. Process mining has been widely used to discover the predefined process model from an event log. The discovered model either shows the connection of reality and original model, or indicates the conformance of the process and the event log, which can be used to discover, monitor and improve the original process. The paper aims to enhance the flexibility and adaptability of the process discovery algorithm. The paper first analyses the process patterns using hierarchical structure, then proposes an improved multi-step process discovery approach based on the first-order Markov transition matrix. Finally, this paper verifies the feasibility and applicability of the improved approach through a simulation example.

Introduction

Information systems such as ERP, CRM and Workflow Management Systems have been used by companies or enterprises to support the execution of their business processes. These information systems typically support the record of event logs which provide detailed information about the execution of processes. One of the purposes of process mining is to discover the process model from an event log. The discovered model either shows the connection of reality and original model, or indicates the conformance of the process and the event log, which can be used to discover, monitor and improve the original process [1]. Current process mining algorithms all based on complete event logs. Process mining can be divided into three types: process discovery, conformance checking and process enhancement. Based on the types of data in an event log, different perspectives of process mining can be discovered. The main three perspectives are control-flow perspective, case perspective and organizational perspective. Process mining is a new application of data mining, which is reproducing the real process of the business through analyzing workflow execution logs [2].

Related Process Mining Algorithms

The idea of process mining first appeared in the field of software engineering, was proposed by Joan than E. Cook in 1995 [3]. In 1998, Agrawal first applied process mining technology to enterprise business process modeling [1]. Based on the mining process, workflow model reconstruction can be divided into single-step reconstruction and multi-step reconstruction [4]. Single-step reconstruction method is directly mined the workflow model based on the activities dependencies which explicit exist in the log. The single-step reconstruction algorithms include directed graph -based mining methods and WF-net based α algorithms proposed by Aalst. α algorithm is one of the classic algorithms of process mining, the mining process is simple, and the computation time is short, but the log noise handling capacity is insufficient, which is not suitable for mining complex structures workflow model such as non-free choice, loop, hidden activities and duplication activities[5]. A lot of research has been done on the improving of α algorithm, such as α* algorithm and β algorithm [6-7]. Multi-step reconstruction method adds the workflow log preprocessing or workflow model evaluation process, which makes the mining with higher accuracy, but the algorithm execution time is longer. Multi-step reconstruction methods include region-based mining method, clustering based
mining method, genetic algorithm based mining method, and frequency/dependency based mining method, multi-model mining and incremental mining methods [8]. These algorithms all required of the integrity of the workflow model, which means that the mined workflow model needs to meet all the log, so the workflow model mined by these algorithms always with lower accuracy.

Process mining has attracted wide attention in academic domain, but there still lack of the algorithms with strong effectiveness for a wide range, good robustness and high efficiency. Some of the algorithms can only identify simple business process structure from the log due to the constraints of formal representation capacity, and some of the mining algorithms have higher requirements of the log, which is difficult to handle the incomplete information or the noise. Paper[9] designed a multi-step process mining method based on the first-order Markov transition matrix which can automatically deduced the basic structures of the relationship between activities. This paper improved the process mining algorithm to establish the actual structure relationship between the activities in order to reconstruct the workflow with higher adaptability and flexibility.

Design of the Improved Process Mining Algorithm

WFMC business divided the process into six basic control structures [10-12], documents [5,7,9] further subdivided the basic six structures into eight kinds(sequential, parallel, synchronization, selection, aggregation, self-cycle, multi cycle I and multi cycle II), this paper makes analysis for the smaller particle size the element flow mode, and adopts the idea of hierarchical representation of loop structure, the task execution sequence cycle as an independent unit, thus stepping multi step cycle can be regarded as special case.

The Preparation and Preprocess of the Log

Workflow log usually records the actual implementation process of the workflow model, and the log usually made up with the workflow instance name ‘Ca_id’, activity name ‘Activity’, performer ‘Performer’ (can be specific people, or the application program), and execution time ‘Time’, etc. Among these, ‘Ca_id’ used to identify execute times of one workflow, such as Ca_1, Ca_2, ..., ‘Activity’ used to identify a specific activity of the workflow process, like x0, x1, ..., ‘Performer’ and ‘Time’ are used to represent the specific actors and execution time of the activity.

Because of the input errors, there may be records missing, duplicate records and other reasons which may cause the noise or workflow logs incomplete. For the log with noise, it can do the noise filtering by setting the frequency threshold ‘θ’. For incomplete logs, it can be filtered using the following two methods: a) list the sets of the end events of the log, if an instance’ send event does not belong to the set, then the instance logs are incomplete and should be removed; b) in the log, if a task is only having the start event without a corresponding end event, or only has the end event without a corresponding start event, then the instance is also incomplete and should be removed.

The Definition of the Logic Relationship Mining Rules

This paper aims to derive the logical relationship in the workflow process through the analysis of the transition matrix P. Xs represents the start of the process,XE represents the end of the process, X represents the sets of process nodes, xi represents the activity node in the process, pij represents the transition possibility between process nodes xi and xj.

The logic relationship mining rules are as follows:

Start node: \( X_s = x_i, \forall x_i \in X, \exists x_j \in X : p_{ij} = 0 \);

End node: \( X_e = x_j, \forall x_j \in X, \exists x_i \in X : p_{ij} = 0 \);

Sequence relationship: \( x_i > x_j, \forall x_i, x_j \in X : p_{ij} \neq 0 \).  

454
And-Split: $x_t$ is the And-Split node, and $x_{n_k} \parallel x_{n_k} \parallel \ldots \parallel x_{n_k}$, if and only if $p_{n_k} = 1$ \& $x_t \neq x_{n_k}$.

And-join: $x_t$ is the And-join node, and $x_{n_k} \parallel x_{n_k} \parallel \ldots \parallel x_{n_k}$, if and only if $p_{n_k} = 1$ \& $x_t = x_{n_k}$.

OR-Split: $x_t$ is the OR-Split node, and $x_{n_k} < x_{n_k} < \ldots < x_{n_k}$, if and only if:

$\exists x_t, x_{n_k} \in S, x_i < x_{n_k}, \ldots < x_{n_k}, x_t \triangleright x_{n_k}, \text{for } t \leq k, t = 0, t + 1$.

OR-join: $x_t$ is the OR-join node, and $x_{n_k} < x_{n_k} < \ldots < x_{n_k}$, if and only if:

$\exists x_t, x_{n_k} \in S, x_t < x_{n_k}, \ldots < x_{n_k} = x_{n_k}$.

Circular relationship: there is a circular between $x_t$ and $x_{n_k}$, if and only if:

$\exists x_t, x_{n_k} \in S, x_t > x_{n_k}$.

a) Start of the multi-step circular $X_{eq}$:

$X_{eq} = x_{n_k}, x_t > x_{eq}$, if and only if $\exists x_{eq} \in X_{eq}, x_t > x_{eq}$.

b) End of the multi-step circular $X_{eq}$:

$X_{eq} = x_{eq} \triangleright x_t$, if and only if $\exists x_{eq} \in X_{eq}, x_t < x_{eq}$.

c) Self-circulation:

$\exists x_t \in X, x_{n_k} = x_{n_k}$, if and only if $p_{tl} = p (p \neq 1)$.

The paper analyzes the cycle structure based on the hierarchical structure, which makes the formal description of the process structure more flexible.

**Design of the Process Mining Algorithm**

Based on the above workflow process logical relationship mining rules, we design the improved workflow mining algorithm $Process (P, X)$. The algorithm expresses the structure relationship in the form of activity relation pairs, and finally establishes three kinds of relation sets: $W$, $W_{and}$, $W_{select}$.

<table>
<thead>
<tr>
<th><strong>The Improved Process Mining Algorithm Process (P, X)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> P,X</td>
</tr>
<tr>
<td><strong>Output:</strong> W, W_{and}, W_{select}</td>
</tr>
<tr>
<td><strong>Initialization:</strong> W = 0, W_{and} = 0, W_{select} = 0</td>
</tr>
<tr>
<td><strong>Step4.1:</strong> Identify the start and end task</td>
</tr>
<tr>
<td>$X_t = x_t \forall x_t \in X, x_t \notin X : p_{tl} = 0$</td>
</tr>
<tr>
<td>$X_t = x_t \forall x_t \in X, x_t \notin X : p_{tl} = 0$</td>
</tr>
<tr>
<td><strong>Step4.2:</strong> Mining the sequence relationship &amp; causal relationship</td>
</tr>
<tr>
<td>for each $(t \leq n + 1, i = 0, i \ldots, f \leq n + 1, f = 0, j = +1)$, do</td>
</tr>
<tr>
<td>$W = W \cup ({x_t, x_j \mid x_t &gt; x_j} \forall x_t, x_j \in X : p_{lj} = 0)$</td>
</tr>
<tr>
<td>$W = W \cup ({x_t, x_j \mid x_t &gt; x_j} \forall x_t, x_j \in X : p_{lj} = 1 &amp; x_t \neq x_j)$</td>
</tr>
<tr>
<td><strong>Step4.3:</strong> Mining the And-Split, And-join, OR-Split, OR-join relationship</td>
</tr>
<tr>
<td>if $\exists x_{n_k}, x_{n_k}, \ldots, x_{n_k} \in X, p_{n_k} = p_{n_k} = \ldots = p_{n_k}$ $= \frac{1}{k} = 0 &amp; x_t = x_{n_k}$</td>
</tr>
</tbody>
</table>

455
Formal Representation of Mining Results

Through the log structure mining algorithm, the structure relations in the process are expressed in the form of relational sets (\(W,\ W_a\) and \(W_{select}\)). On the basis of the three relation sets, a variety of formal description languages can be used to establish the corresponding process model. Refer to the \(a\)-algorithm, the corresponding \(WF\)-net model can be generated.

Case Study

To verify the feasibility of the approach, this paper chooses a business process workflow management system logs generated by Staff ware (use ‘Pro’ as short in the below) as the simulation example. The log file is get from the W.M.P. van der Aalst team open database [2]. The predefined model of the process is shown in Fig. 1.
After the process ‘Pro’ running for some time, the modelers collected the workflow log from the enterprise WFMS (see Table 1).

Table 1. Process event log.

<table>
<thead>
<tr>
<th>Ca_id</th>
<th>Ac</th>
<th>Ca_id</th>
<th>Ac</th>
<th>Ca_id</th>
<th>Ac</th>
<th>Ca_id</th>
<th>Ac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ca_1</td>
<td>A</td>
<td>Ca_2</td>
<td>D</td>
<td>Ca_2</td>
<td>E</td>
<td>Ca_2</td>
<td>H</td>
</tr>
<tr>
<td>Ca_1</td>
<td>B</td>
<td>Ca_3</td>
<td>D</td>
<td>Ca_1</td>
<td>H</td>
<td>Ca_4</td>
<td>E</td>
</tr>
<tr>
<td>Ca_2</td>
<td>A</td>
<td>Ca_1</td>
<td>F</td>
<td>Ca_1</td>
<td>G</td>
<td>Ca_3</td>
<td>G</td>
</tr>
<tr>
<td>Ca_3</td>
<td>A</td>
<td>Ca_1</td>
<td>F</td>
<td>Ca_3</td>
<td>E</td>
<td>Ca_4</td>
<td>F</td>
</tr>
<tr>
<td>Ca_1</td>
<td>C</td>
<td>Ca_4</td>
<td>B</td>
<td>Ca_2</td>
<td>F</td>
<td>Ca_3</td>
<td>E</td>
</tr>
<tr>
<td>Ca_3</td>
<td>B</td>
<td>Ca_5</td>
<td>A</td>
<td>Ca_3</td>
<td>F</td>
<td>Ca_4</td>
<td>G</td>
</tr>
<tr>
<td>Ca_1</td>
<td>D</td>
<td>Ca_2</td>
<td>C</td>
<td>Ca_4</td>
<td>D</td>
<td>Ca_4</td>
<td>H</td>
</tr>
<tr>
<td>Ca_2</td>
<td>B</td>
<td>Ca_3</td>
<td>C</td>
<td>Ca_2</td>
<td>G</td>
<td>Ca_3</td>
<td>G</td>
</tr>
<tr>
<td>Ca_4</td>
<td>A</td>
<td>Ca_4</td>
<td>C</td>
<td>Ca_2</td>
<td>E</td>
<td>Ca_4</td>
<td>H</td>
</tr>
<tr>
<td>Ca_1</td>
<td>E</td>
<td>Ca_1</td>
<td>G</td>
<td>Ca_3</td>
<td>F</td>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>

(PS: Table 1 only lists the Ca_1-Ca_4 considering the article length restriction)

Based on the log, we can get the Markov transition matrix as follows [9]:

\[
P = \begin{bmatrix}
0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.15 & 0.85 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.47 & 0.53 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(1)

Execute the process mining algorithm Process \((P,X)\), \(X=(A,B,C,D,E,F,G,H)\), we can get the logical relationship sets \(W\), \(W_{\text{and}}\), and \(W_{\text{select}}\):

\[
\]

\[
W_{\text{select}} = \{(B, C)(E, F)(F, G)\}.
\]

According to the relationship sets \((W, W_{\text{and}} \text{ and } W_{\text{select}})\) and the activity set \(X\), the reconstructed model of ‘Pro’ is shown in Fig.2.

![Figure 2. The reconstructed model of ‘Pro’.](image)

**Analyze:** The resulting WF-net is same as the previously designed model (Fig.1). Through the above experiments, the feasibility of the improved method has been verified.
Summary

Based on complete event log, the first-order Markov transition matrix of the log was established, and the ordering relations identification rules has been defined, based on these, the structures mining algorithm has been developed. Furthermore, the proposed algorithm represents the cyclic structure using hierarchical structure which improved the efficiency and the adaptability of the original algorithm.

Acknowledgement

This research was financially supported by the Southwest Forestry University Research Startup Fund Project.

References


