www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 9 Sep 2015, Page No. 14383-14386

Extension to Alpha Algorithm for Process Mining

Anoopam Banerjee[1], Preeti Gupta[2]

[1]Amity School of Engineering & Technology, Amity University Rajasthan,
Jaipur, Rajasthan 303002, India
anoopambanerjee@outlook.com

[2]Amity School of Engineering & Technology, Amity University Rajasthan,
Jaipur, Rajasthan 303002, India
pgupta@jpr.amity.edu

Abstract: ***Different Enterprise Information Systems store relevant information about the events occurring in the enterprise. The information stored may be of data, process, etc. and is well structured. The stored information paves the path for Process Mining. Many approaches are suggested for Process Mining, out of which the core approach is algorithmic approach. In Process Mining, Alpha Algorithm assumes importance as it aims at reconstructing causality, from a set of sequences of events that take place in an organization. The paper focuses on identifying limitations of Alpha Algorithm and giving an extension to increase its functionality.***

Keywords: Process Mining, Alpha Algorithm, Petri Nets, Workflow Log, Event Log.

1. Introduction

Gathering information about the processes as they take place (logs) is the first step to initiate business process mining which is in total contrast to start with a process design. This information can be retrieved in some form like spreadsheet or transactional logs from transactional systems like Customer Relationship Management (CRM), Enterprise Resource & Planning (ERP) or Work Flow Management (WFM) systems. While retrieving this information one important assumption made is that each and every events of a process in terms of the occurrence of the events is available. This assumption will validate the process model for its completeness, as all the events are recorded for that process. The goal of process mining is to extract information about processes from transaction logs [1].

There are three possible methods of process discovery [2]. One using neural networks, second is pure algorithmic approach and third is Markovian approach. Out of the above three later two are considered as most promising. The pure algorithmic approach, builds an FSM where states are fused together if their future is identical. The markovian approach combines algorithmic approach with statistical methods thus deals with noise (in appropriate data).

Here pure algorithmic approach is used to build a process model that describes itself in terms of process with originator and time as an additional feature. The base algorithm considered is Alpha Algorithm (α-algorithm). Additional five steps are added to this algorithm to make it more versatile and helps to analyze the process in an efficient manner.

2. Process Mining using Alpha Algorithm

To illustrate the example of process mining, a sample event log is taken into consideration as shown in table 1 below. Using the same event log, the extension to the Alpha Algorithm will be shown.

Table 1: A sample event log

Case Id	*Activity Id*	*Originator*	*Start Timestamp*	*End Timestamp*
Case 1	Activity A	John	9-3-2004:15.01	9-3-2004:15.09
Case 2	Activity A	John	9-3-2004:15.12	9-3-2004:15.45
Case 3	Activity A	Sue	9-3-2004:16.03	9-3-2004:16.05
Case 3	Activity B	Carol	9-3-2004:16.07	9-3-2004:16.59
Case 1	Activity B	Mike	9-3-2004:18.25	9-3-2004:19.20
Case 1	Activity C	John	10-3-2004:9.23	10-3-2004:10.22
Case 2	Activity C	Mike	10-3-2004:10.34	10-3-2004:11.35
Case 4	Activity A	Sue	10-3-2004:10.35	10-3-2004:11.05
Case 2	Activity B	John	10-3-2004:12.34	10-3-2004:12.42
Case 2	Activity D	Pete	10-3-2004:12.50	10-3-2004:13.10
Case 5	Activity A	Sue	10-3-2004:13.05	10-3-2004:14.21
Case 4	Activity C	Carol	11-3-2004:10.12	11-3-2004:11.19
Case 1	Activity D	Pete	11-3-2004:10.14	11-3-2004:10.57
Case 3	Activity C	Sue	11-3-2004:10.44	11-3-2004:10.50
Case 3	Activity D	Pete	11-3-2004:11.03	11-3-2004:11.49
Case 4	Activity B	Sue	14-3-2004:11.18	14-3-2004:12.27
Case 5	Activity E	Carol	17-3-2004:12.22	17-3-2004:13.41
Case 5	Activity D	Carol	18-3-2004:14.34	18-3-2004:15.31
Case 4	Activity D	Pete	19-3-2004:15.56	19-3-2004:16.32

To start with process mining, the above log serves as the step one. There are basically three different perspective for process mining. These are:

a) The process perspective ("How?").
b) The organizational perspective ("Who?")

c) The case perspective ("What?")

The main goal of process perspective is to focus on the control flow of the entire process by showing the steps included in the process in a sequential manner. The entire ordering of the events are shown with the help of Petri Nets [3] or Event Driven Process Chains [4].

Organizational perspective targets to the originator field from the log table. It shows that which person is involved in that process and how they are related. The result of this perspective is to differentiate people in terms of the organizational structure i.e. roles and responsibilities [5, 6, 7, 8]. This paper will show how to merge organizational perspective with the process perspective.

The case perspective focuses on properties of the case. Cases can be characterized by their path in the process or by the originators working on a case. However, cases can also be characterized by the values of the corresponding data elements.

Alpha Algorithm is one of the basic 8-step algorithm that is used in process mining. It is aimed at reconstructing causality from a set of sequences of events. It was first put forward by van der Aalst, Weijters and Măruşter [9].

Petri Nets [3] (also known as P/T nets) with special properties (workflow nets) is produced as the output of this algorithm from event logs. Any log generating business management tool like an ERP system, etc. provides the event logs required. The net consists of events and transitions from events i.e. the task being performed over the events. The entire petri net depicts the entire process as a process model that is recorded in that log. The input for the algorithm is a workflow log $W \subseteq T^*$ and a workflow net being constructed as the output of the algorithm, where T is the set of all the task under consideration. The net is prepared by examining various relationships that is observed between the tasks recorded in the log. The relationship focused is of causality. The other relationships derived and used are direct succession, parallel and choice. Causality can be explained as one specific task might always precede another specific task in every execution trace/log. Mathematically the relations are deduced as:

- Direct succession: $a > b$; if and only if for some case, a is directly followed by b.
- Causality: $a \rightarrow b$; if and only if for some case, $a>b$ and not $b>a$
- Parallel: $a \parallel b$; if and only if for some case, $a>b$ and $b>a$
- Choice: $a \# b$; if and only if not $a>b$ and not $b>a$

The 8 steps Alpha Algorithm (α-algorithm) is stated as under [10]:

Let W be a workflow log over T. α(W) is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\ t \in \sigma \}$

T_W is a set of all tasks which occur in at least one trace.

2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\ t = first(\sigma) \}$

T_I is a set of tasks which occur in trace initially

3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\ t = last(\sigma) \}$

T_O is a set of task which occur in trace terminally

4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A \forall b \in B}\ a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\ a1 \#_W a2 \wedge \forall_{b1,b2 \in B}\ b1 \#_W b2 \}$

X_W is a set of all pairs in which places are discovered.

5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$

Y_W is a set in which places are identified as pair of set of task for minimal places.

6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_w, o_w\}$

P_W is the set that contains one place $p_{(A,B)}$ for each pair in Y_W with input place i_w and output place o_w.

7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_w, t) \mid t \in T_I \} \cup \{ (t, o_w) \mid t \in T_O \}$

F_W is the set defining the flow relation.

8. $\alpha(W) = (P_W, T_W, F_W)$.

Here α(W) is the process model

Considering the above algorithm and applying it on the event log from Table 1,

$W = \{<ABCD>^2, <ACBD>^2, <AED>^1\}$

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\ t \in \sigma \}$
   $T_W = \{A, B, C, D, E\}$

2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\ t = first(\sigma) \}$
   $T_I = \{A\}$

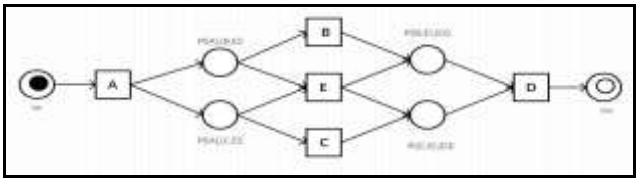
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\ t = last(\sigma) \}$
   $T_O = \{D\}$

4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A \forall b \in B}\ a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\ a1 \#_W a2 \wedge \forall_{b1,b2 \in B}\ b1 \#_W b2 \}$
   $X_W = \{(\{A\},\{B\}), (\{A\},\{C\}), (\{A\},\{E\}), (\{B\},\{D\}), (\{C\},\{D\}), (\{E\},\{D\}), (\{A\},\{B,E\}), (\{A\},\{C,E\}), (\{B,E\},\{D\}), (\{C,E\},\{D\})\}$

5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\ A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$
   $Y_W = \{(\{A\},\{B,E\}), (\{A\},\{C,E\}), (\{B,E\},\{D\}), (\{C,E\},\{D\})\}$

6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_w, o_w\}$
   $P_W = \{p_{(\{A\},\{B,E\})}, p_{(\{A\},\{C,E\})}, p_{(\{B,E\},\{D\})}, p_{(\{C,E\},\{D\})}, i_w, o_w\}$

7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_w, t) \mid t \in T_I \} \cup \{ (t, o_w) \mid t \in T_O \}$
   $F_W = \{(i_w,\{A\}), (\{A\}, p_{(\{A\},\{B,E\})}), (\{A\}, p_{(\{A\},\{C,E\})}), (p_{(\{A\},\{B,E\})},\{B\}), (p_{(\{A\},\{B,E\})},\{E\}), (p_{(\{A\},\{C,E\})},\{C\}), (p_{(\{A\},\{C,E\})},\{E\}), (\{B\},p_{(\{B,E\},\{D\})}), (\{E\}, p_{(\{B,E\},\{D\})}), (\{C\},p_{(\{C,E\},\{D\})}), (\{E\},p_{(\{C,E\},\{D\})}), (p_{(\{B,E\},\{D\})},\{D\}), (p_{(\{C,E\},\{D\})},\{D\}), (\{D\}, o_w) \}$



8. $\alpha(W) = (P_W, T_W, F_W)$

Figure 1: Process model generated from process log using Alpha Algorithm

The process model generated using Alpha Algorithm is shown in Figure 1 above.

This algorithm is basically of theoretical interest. Also it is

too simple to be applicable to real life logs. However it is the basic algorithm that discovers the process from a log and generates a workflow to deduce the current ongoing process in just 8 lines.

The demerits of the algorithm include its limited response against the log that only shows the process flow. This algorithm do not attempt to solve any of the major challenges which includes mining hidden task (hidden processes inside a process), mining duplicate task (due to re-entry of data in log), mining different perspective (process, organizational and case), dealing with noisy data (improper / corrupt data present in the log), time constraint (using timestamp factor present in log), mining loops, etc.

3. Proposed Methodology: Extension to the Alpha Algorithm ($\alpha^{\#}$ Algorithm)

The following five steps are proposed as an extension to the original Alpha Algorithm:

9. $M_W = \{(T_{min}, T_{avg}, T_{max}, a) \mid \exists_{\sigma \in W}\ a \in \sigma \wedge T_{min} = \min(etime(a) - stime(a)) \wedge T_{avg} = avg(etime(a) - stime(a)) \wedge T_{max} = \max(etime(a) - stime(a))\}$

M_W gets the minimum (Tmin), average (Tavg) and maximum (Tmax) time for each event(a) that is in log. etime is the end time for an event and stime is the start time for the event.

10. $Z_W = \{(A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A \wedge b \in B}\ a \rightarrow_W b \wedge \forall_{a \in A \wedge b \in B}\ a \|_W b \wedge \forall_{b \in A \wedge a \in B}\ b \|_W a\}$

Z_W is a set of all pairs which shows a transition in a process

11. $N_W = \{(T_{min}, T_{avg}, T_{max}, (A,B)) \mid (A,B) \in Z_W \wedge T_{min} = \min(stime(B) - etime(A)) \wedge T_{avg} = avg(stime(B) - etime(A)) \wedge T_{max} = \max(stime(B) - etime(A))\}$

N_W gets the minimum (Tmin), average (Tavg) and maximum (Tmax) time for each transition ($p_{(A,B)}$) between A to B that is in log. etime is the end time for an event and stime is the start time for the event.

12. $O_W = \{(o,a) \mid \exists_{\sigma \in W}\ a \in \sigma \wedge o = originator(a)\}$

Ow gets the originator (o) who performed the event/task (a) in the log.

13. $\alpha^{\#}(W) = \{\alpha(W), M_W, N_W, O_W\}$

$\alpha^{\#}(W)$ is the new process model which extends the current α algorithm

Final step i.e. 8th step of the Alpha Algorithm, gave the process perspective model of the sample log from Table 1. The inclusion of the 9th and 11th step gives the generated process model a new dimension of time which helps to analyse the model in more depth and identify the issues relating to delay in the process. This will serve as a solution to a major challenge with this algorithm i.e. the time constraint. Now the algorithm is using the time constraint as a major factor in determining the time consumed by events and the transitions between the events. After including the 12th step, the response of the algorithm widens to organizational perspective as it shows the originator performing the respective tasks. 10th step is a supporting step for the 11th step which retrieves the transition in the process model.

After applying the new steps to the event log,

9. $M_W = \{(T_{min}, T_{avg}, T_{max}, a) \mid \exists_{\sigma \in W}\ a \in \sigma \wedge T_{min} = \min(etime(a)-stime(a)) \wedge T_{avg} = avg(etime(a)-stime(a)) \wedge T_{max} = \max(etime(a)-stime(a))\}$

M_W = {(2,29.8,76,{A}), (8,46,69,{B}), (6,48.25,67,{C}), (20,40.4,57,{D}), (79,79,79,{E})}

10. $Z_W = \{(A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A}\ \forall_{b \in B}\ a \rightarrow_W b \wedge \forall_{a \in A}\ \forall_{b \in B}\ a \|_W b \wedge \forall_{b \in A}\ \forall_{a \in B}\ b \|_W a\}$

Z_W = {({A},{B}), ({B},{C}), ({C},{D}), ({A},{C}), ({C},{B}), ({B},{D}), ({A},{E}), ({E},{D})}

11. $N_W = \{(T_{min}, T_{avg}, T_{max}, (A,B)) \mid (A,B) \in Z_W \wedge T_{min} = \min(stime(B) - etime(A)) \wedge T_{avg} = avg(stime(B) - etime(A))\}$

N_W ={(0.02,1.59,3.16,({A},{B})), (14.03,15.74,17.45,({B},{C})), (0.13,11.825,23.52,({C},{D})), (18.49,20.78,23.07,({A},{C})), (0.59,36.09,71.59,({C},{B})), (0.08,61.685,123.29,({B},{D})), (166.01,166.01,166.01,({A},{E})), (24.53,24.53,24.53,({E},{D})) }

12. $O_W = \{(o,a) \mid \exists_{\sigma \in W}\ a \in \sigma \wedge o = originator(a)\}$

Ow = {((John, Sue),{A}), ((Carol, John, Mike, Sue),{B}),((Carol, John, Mike, Sue),{C}), ((Pete, Carol),{D}), ((Carol),{E})}

13. $\alpha^{\#}(W) = \{\alpha(W), M_W, N_W, O_W\}$

After executing 13th step, some data will be included in the petri net as generated in figure 1. The first set of data is shown in figure 2 below:

Activityid	Originator(s)	Min Time (Min)	Avg Time (Min)	Max Time (Min)
Activity A	John, Sue	2	29.8	76
Activity B	Carol, Mike, John, Sue	8	46	69
Activity C	John, Mike, Carol, Sue	6	48.25	67
Activity D	Pete, Carol	20	40.4	57
Activity E	Carol	79	79	79

Figure 2: Data generated from process log after using step 9(M_W) and 12(O_W) of $\alpha^{\#}$algorithm

The Column "Originator(s)", in figure 2, shows the person(s) who performed the corresponding activity during the whole process. This output shows the organizational perspective of a process model.

Similarly the columns, "Min Time (Min)", "Avg Time (Min)" and "Max Time (Min)", in figure 2, shows the minimum, average and maximum time taken by the activity/task (here in minutes), respectively. Here the time parameter is used to check for any delay in particular events. The minimum, average and maximum time will give an overview to the user about the time consumed by events in the current process scenario.

The second set of data is shown in the figure 3 below:

Transition		Min Time (in Hr)	Avg Time (in Hr)	Max Time (in Hr)
From	To			
Activity A	Activity B	0.02	1.59	3.16
Activity B	Activity C	14.03	15.74	17.45
Activity C	Activity D	0.13	11.825	23.52
Activity A	Activity C	18.49	20.78	23.07
Activity C	Activity B	0.59	36.09	71.59
Activity B	Activity D	0.08	61.685	123.29
Activity A	Activity E	166.01	166.01	166.01
Activity E	Activity D	24.53	24.53	24.53

Figure 3: Data generated from process log after using step 10(Z_W) and 11(M_W) of $\alpha^{\#}$algorithm

The columns "Min Time (in Hr)", "Avg Time (in Hr)", and "Max Time (in Hr)", in figure 3, shows the minimum, average and maximum time taken by the corresponding transition as logged in the workflow log, respectively. Here the time parameter is used to check for any delay in transition between events. The minimum, average and maximum time will give an overview to the user about the time consumed between the events in the current process scenario.

4. Conclusion and Future Work

Through this paper, business process mining and implementation of Alpha Algorithm on a simple process log was seen. Based on the process log and Alpha Algorithm, a process model was built which demonstrated the process perspective with the help of Petri-Nets. Five steps were included to the existing Alpha Algorithm to widen the scope of model generated by including the organizational perspective and eliminating one of the major issue i.e. using time constraint in process mining.

As a part of future work, to give a more precise view to the process model generated, noisy data should be removed from the log. To achieve this an inclusion of a parameter, say EoP (End of Process), can be made in the log to identify the completed processes. Inclusion of this parameter will constrain the Alpha Algorithm to fetch minimum data from log which will result in the accurate modelling of the process. This will enhance the organizational perspective of the model generated. Also the time parameter can be expanded to get time of each workflow log individually.

References

[1] Van der Aalst, Wil MP, and A. J. M. M. Weijters. "Process mining: a research agenda." *Computers in industry* 53.3 (2004): 231-244.

[2] Cook, Jonathan E., and Alexander L. Wolf. "Discovering models of software processes from event-based data." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 7.3 (1998): 215-249.

[3] Reisig, Wolfgang, and Grzegorz Rozenberg. *Lectures on petri nets i: basic models: advances in petri nets*. Vol. 1491. Springer Science & Business Media, 1998.

[4] Keller, Gerhard, and Thomas Teufel. *SAP R/3 process oriented implementation*. Addison-Wesley Longman Publishing Co., Inc., 1998.

[5] Nemati, Hamid R., and Christopher D. Barko, eds. *Organizational data mining: leveraging enterprise data resources for optimal performance*. IGI Global, 2004.

[6] Moreno, Jacob Levy. "Who shall survive?: A new approach to the problem of human interrelations." (1934).

[7] Scott, John. *Social network analysis*. Sage, 2012.

[8] Wasserman, Stanley, and Katherine Faust. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.

[9] Van der Aalst, Wil, Ton Weijters, and Laura Maruster. "Workflow mining: Discovering process models from event logs." *Knowledge and Data Engineering, IEEE Transactions on* 16.9 (2004): 1128-1142.

[10] Prof. Dr. Ir. Wil van der Aalst, Process Mining: Beyond Business Intelligence, http://www.processmining.org/_media/presentations/processminingtutorialesscass-2009.pdf

Author Profile

Anoopam Banerjee is a research scholar in the Department of Computer Science & Engineering in Amity School of Engineering & Technology, Amity University Rajasthan. He is having research interest in the field of Data Mining and Software Testing.

Preeti Gupta is associated with Amity University Rajasthan (Department of Computer Science & Engineering) as Assistant. Professor. Her research interest comprises of Data Mining and Knowledge Management.