Journal of Computers Vol. 30 No. 4, 2019, pp. 140-150
doi:10.3966/199115992019083004013

A Novel Approach to Discover Precise Process Model by Filtering out Log Chaotic Activities

Guo Yi[1], Zhang Peng[1*]

[1] College of Computer Science and Engineering Shandong University of Science and Technology, Qingdao, Shandong 266590, China
{ygcrawler, bigbigroc}@163.com

Received 13 January 2019; Revised 2 March 2019; Accepted 12 March 2019

Abstract. Process mining technology automatically discovers business processes from the execution data of business processes. The real life business process event data logs usually contain chaotic activities which make the traditional event data log filtering approach not able to effectively filter the chaotic activities in event data logs. This paper proposes a novel chaotic activities filtering approach based on bidirectional causal dependence. The approach achieves the filtering of chaotic activities in event data logs by analyzing the bidirectional causal dependence between the model and event data logs and taking the precision as a constraint. At the end of this paper, the proposed approach is used in the Tianyuan big data platform to verify the effectiveness. By comparison experiments of chaotic activity filtering approach based on information entropy is evaluated from the aspects of time complexity. The evaluation shows that the approach can discover more precise process models through the analysis of precision between multiple sets of event data logs and the process models generated before and after chaotic activities filtering.

Keywords: bidirectional causal dependence, chaotic activities filtering, precision, process mining

1 Introduction

Business process management (e.g. [1-4]) affects the operational efficiency and quality of business processes with the emergence of new business processes and changes in existing business processes. Reasonably analyzing the event logs that record the execution of business process activities, and then automatically discovering accurate business processes is the premise of business process auditing, analysis, and enhancement. It is also the key to ensure the operational efficiency and quality of business process.

Process mining (e.g. [5-8]) is a cross discipline in the data mining field and process modeling and analysis field, and it focuses on analyzing the event log during the execution of business processes and the process of automatic modeling. Event logs can be used to conduct three types of process mining: discovery, conformance check and enhancement. Process discovery, which plays a prominent role in process mining, is the task of automatically generating a process model that accurately describes a business process based on such event data. Many process discovery techniques have been developed over the last decade (e.g. [9, 11-14]), producing process models in various forms, such as Petri nets, process trees, and BPMN models. Starting point of process mining is an event log, and the quality of event logs always affects the quality of discovery models. Real life events logs often contain all sorts of data quality issues [15], include incorrectly logged events, events that are logged in the wrong order, and events that took place without being logged. Instances of such data quality issues are often referred to as noise. Chaotic activities in event logs are a kind of noise, which is relatively independent of the process and can occur at any point in the process. Because of the randomness of chaotic activities in the event log, so effectively identifying chaotic activities is very important for finding accurate process models.

* Corresponding Author

The current modus operandi for filtering activities from event logs is to simply filter out infrequent activities. The plugin 'Filter Log using Simple Heuristics' in the ProM process mining toolkit [16] offers tool support for infrequent activities filtering. The fewer occurrences of an activity in the log, the less likely the activity is to establish relationships with other activities and the less frequent the activity. Conforti et al. [17] recently proposed an approach to filter out outlier events from an event log. The approach first builds an event log prefix automaton to eliminate infrequent arcs. And finally, the events belonging to removed arcs are filtered out from the event log. From the definition of chaotic activity, it can be known that chaos activity is not associated with infrequent degrees, and because of the randomness of chaotic activity, chaotic activity can appear frequently or infrequently in the event log. Ghionna et al. [18] proposed an approach to identify outlier traces from the event log that based on clustering similarity between different traces. The number of traces used for model discovery obtained by this approach will be reduced when chaotic activity exists in most of traces, which will eventually fail to find a precision process model. Tax N et al. [19] proposed an approach to filter chaotic activities based on Information entropy, and the entropy calculation process in this approach is complicated and time-consuming when the number of traces and event types contained in the event log is large.

In this paper, we show that existing approaches do not solve the problem of chaotic activities and we present a chaotic activities filtering approach based on bidirectional causal dependence. With the help of precision (e.g. [20-22]) between model and event log, the approach effectively filters the chaotic activities in event logs based on analyzing the bidirectional causal dependence between activities in event logs. At the end of this paper, the proposed approach is used in the Tianyuan big data platform to verify the effectiveness.

2 Relevant Concepts

Some basic definitions and notations are introduced in this section. For example, the definitions of Petri nets, event log, bidirectional causal dependence and so on. Petri nets can be used for the modeling and analysis of distributed concurrent systems. Petri nets can describe the structure of systems, and can simulate the operation of systems.

Definition 1. (*Multisets and sequence*)

Let A be a set, we denote the mapping $B: A \to B(A)$ where $B(A)$ represents the set of all multisets on set A. For example, B_1=[], B_2=[a], B_3=[a, b, b, c, b, a] and B_4=[a^2, c, b, b^2] are multisets over $A=\{a,b,c\}$. We denote sequence $\sigma=<a_1,a_2,a_n>\in A^*$ where a_i belongs to A and n is a positive integer, $|\sigma|$ indicates the length of the sequence σ and $\sigma(i)$ indicates the value of the i-th element in the sequence σ. The empty sequence is denoted as <>.

Definition 2. (*Petri Net*)

Petri net is a model used to describe distributed concurrent systems. A Petri net is a triple $N=(P,T,F)$, where P and T are finite disjoint sets of places and transitions, respectively, and $F \subseteq (P\times T)\bigcup(T\times P)$ is the flow relation. Let $N=(P,T,F)$ is a Petri net, we denote $x=\{y \mid y\in P\bigcup T \wedge (y,x)\in F\}$ as the pre-set of x and $x=\{y \mid y\in P\bigcup T \wedge (x,y)\in F\}$ as the post-set of x where x belongs to $P\bigcup T$.

Definition 3. (*Marked Petri Net*)

A Marked Petri net is a quadruple $N=(P,T,F,M)$, where $M: P\to\{0,1,2,...\}$ is a mark known as Petri net. A Marked Petri net has an initial mark, denoted as M_0. There are the following transition firing rules of the Marked Petri net：

(1) For the transition $t\in T$, if $\forall p\in P: p\in t \to M(p)\geq 1$, then the transition t may be fired in the mark M, denoted as $M[t>$.

(2) $M[t>$ indicates that the transition t can fire under the mark M, and a new mark M' is obtained due to the fire of the transition t (denoted as $M[t>M'$)

For $\forall p\in P$, the transition firing rule can expressed as follows:

$$M'(p)=\begin{cases} M(p)-1: p\in \cdot t - t\cdot \\ M(p)+1: p\in t\cdot - \cdot t \\ M(p), else \end{cases}$$

Definition 4. (*Event Logs*)
The event log is composed of cases and the case is composed of events. The events in the case are represented in the form of trace, which can also be regarded as the sequence of events. Given set A represents a collection of all labels in the event log and each label in the trace represents an event. For example L=[<*a*, *c*, *d*, *h*>, <*a*, *b*, *c*, *d*, *f*, *g*>, <*a*, *c*, *d*, *e*, *f*, *g*>]. Given a Petri net as $N=(P,T,F,M)$, if there is no corresponding label in transitions of net *N*, then denoted as $\lambda(t)=\tau$, where transition t becomes an invisible transition and λ represents the mapping between labels and transitions. $\exists t\in T, \lambda: t\to A\bigcup\{\tau\}$, if $\lambda(t)\neq\tau$, then $\lambda(t)\in A$.

Definition 5. (*Basic Sequence Relationship of Events*)
Given event log L based on A, and $a>_L b$, where $a,b\in A$, indicates that there exists a trace $\sigma=<t_1,t_2,...,t_n>, i\in\{1,...,n-1\}$ where $\sigma\in L$, $\lambda(t_i)=a$ and $\lambda(t_{i+1})=b$. The formula $a\to_L b$ is unidirectional causal dependence between events if and only if and only if formula $a>_L b$ and formula $b\not>_L a$ are established.

3 Analysis of Chaotic Activities

The real business process event log usually contains chaotic activities. The chaotic activity is independent for the entire process and can occur at random time points. The existing filtering approaches of chaotic activities cannot effectively find and filter chaotic activities when chaotic activity is contained in most of the trace of the event log. As a result, we cannot find accurate business process models and the entire business process cannot be properly audited, analyzed, and improved. In this section we propose the concept of bidirectional causal dependence, based on which we finally find and filter the chaotic activity in the event log.

The discovery of chaotic activity takes the compensation request process as an example. The process contains 8 steps: (A) register request, (B) examine thoroughly, (C) examine casually, (D)check ticket, (E) decide, (F) re-initiate request, (G) pay compensation, and (H)reject request. Event log L contains the 6 traces, L=[<*A*, *B*, *D*, *E*, *H*>, <*A*, *D*, *C*, *E*, *G*>, <*A*, *C*, *D*, *E*, *F*, *B*, *D*, *E*, *G*>, <*A*, *D*, *B*, *E*, *H*>, <*A*, *C*, *D*, *E*, *F*, *D*, *C*, *E*, *F*, *C*, *D*, *E*, *H*>, <*A*, *C*, *D*, *E*, *G*>].We found the process model shown in Fig. 1 based on the Inductive Miner algorithm.

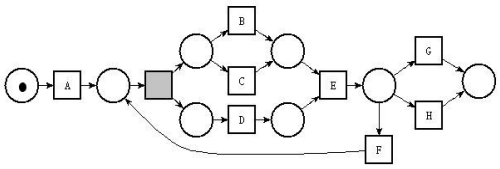

Fig. 1. Process model based on event data Log *L*

The activities that can occur spontaneously at random times are called chaotic activities. The chaotic activity affects the quality of the entire model discovery results. Activity *X* is a customer making a call at any point in time, and customers can make multiple calls at random times. Table 1 shows the event log L' obtained after adding the activity *X* to the event log *L* and Fig. 2 shows the process model discovered by the Inductive Miner algorithm based on the process event log in Table 1.

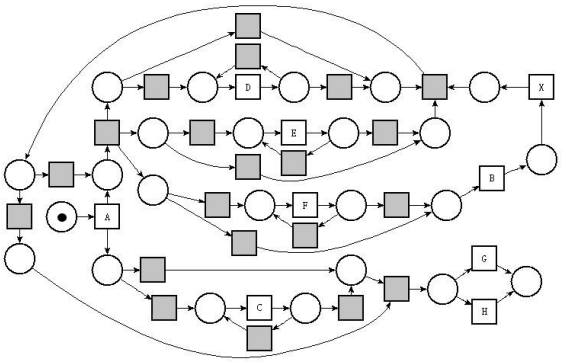

Fig. 2. Process model based on event data Log L'

Table 1. Event Log L'

Event Sequences
$< X, A, B, D, E, H >$
$< A, D, C, E, G >$
$< A, X, C, D, E, F, B, D, X, E, G >$
$< A, D, B, E, H >$
$< A, C, D, E, F, D, C, E, F, C, D, E, X, H >$
$< A, C, X, D, E, G >$

For the traces in the event log contained in Table 2, we use existing methods based on filtering infrequent activities to find the probability of each activity appearing in the event log. $p(A)$ is the probability of activity A appearing in the event log, where $p(A) \approx 0.1277$, and we can also calculate the probability of other activities appearing in the event log: $p(B) \approx 0.0638$, $p(C) \approx 0.1277, p(D) \approx 0.1915$, $p(E) \approx 0.1915, p(F) \approx 0.0638, p(G) \approx 0.0638, p(H) \approx 0.0638, p(X) \approx 0.1064$. We can note that activities F, G and H have the lowest probability in the event log, and activity X is not included in the activity set with the lowest probability. Therefore, the filtering approach based on infrequent activities cannot solve the problem of chaotic activities.

Table 2. Two-way causal dependence

Activity	bidirectional causal dependency sets	Activity	bidirectional causal dependency sets
A	$\{X\}$	F	$\{\}$
B	$\{D\}$	G	$\{\}$
C	$\{D,X\}$	H	$\{\}$
D	$\{B,C,X\}$	X	$\{A,C,D,E\}$
E	$\{X\}$		

From Fig. 2, we can see that due to the appearance of chaotic activity X, the discovery model is too complex and there is a large amount of invisible transitions in the process of model discovery. Based on the relationship between activities from the perspective of Fig. 2, we can see that there is no one-way causal dependency between activity X and some activities, such as $X >_L A, A >_L C, C >_L X$. Therefore, this section proposes the concept of bidirectional causal dependence and bidirectional causal dependence activity set for the discovery of chaotic activity.

Definition 6. (*Bidirectional Causal Dependence*)

Given $a \leftrightarrow_L b$ a bidirectional causal dependence between events where a,b belong to activity set A if and only if $a >_L b, b >_L a$. Based on the definition of the bidirectional causal dependence, we can obtain the bidirectional causal dependence in the process model of Fig. 2 as follows. $A \leftrightarrow_L X, B \leftrightarrow_L D$, $C \leftrightarrow_L D, C \leftrightarrow_L X, D \leftrightarrow_L X, E \leftrightarrow_L X$.

Definition 7. (*bidirectional causal dependence activity sets* BcS_T)

Given a Marked Petri net $N=(P,T,F,M)$ and A-based event log L to define bidirectional causal dependence activity sets for transition t. $BcS_T=\{t_i \mid \exists t_i \in T : l(t) \leftrightarrow_L l(t_i)\}, t \in T$. According to the definition of bidirectional causal dependence activity sets, we get the bidirectional causal dependency activity sets for each activity in the process model shown in Fig. 2, and then get the table of bidirectional causal dependency sets shown in Table 2.

4 Chaotic Activities Filtering Approach Based on Bidirectional Causal Dependence

In the filtering of chaotic activities, the existing event log filtering approach based on infrequent activities cannot solve the problem of chaotic activity filtering. We propose a bidirectional causal dependence as a precondition for filtering chaotic activities. This chapter gives the definition of the bidirectional causal dependence probability, and proposes a chaotic activity filtering approach based on the bidirectional causal dependence, and finally performs the specific chaotic activity filtering operation.

Definition 8. (*bidirectional causal dependence probability* P_a^{Bcd})

Given a Marked Petri net $N=(P,T,F,M)$ and A-based event log L, we give the definition of the bidirectional causal dependence probability of activity a shown as P_a^{Bcd}. Existence $P_a^{Bcd}=|BcS_a|/|A|$ where $a \in A, t \in T$ and $A=\lambda(t)$.

In order to perform the chaotic activity filtering operation, we propose a chaotic activity filtering algorithm based on bidirectional causal dependence according to the above definition. The following is a specific chaotic activity filtering algorithm:

4.1 Algorithm: Chaotic Activity Filtering Algorithm Based on Bidirectional Causal Dependence

Input : Initial Event Log L , Activity Set $\mathcal{A}$

Output : Event Log L'

$L'=\varphi, S_{UC}=\varphi$ //S_{UC} is the unidirectional causal dependence set

$S_p=\varphi$ //S_p is the set of bidirectional causal dependence

for each a in $\mathcal{A}$ do

  $BcS_a=\{\}$ //BcS_a is activity a bidirectional causal dependence set

for each σ_i in L do

  for each t_j in σ_i do

    if $t_{j+1} \rightarrow t_j$ not in S_{UC} then

      $t_{j+1} \rightarrow t_j$ add to S_{UC}

    else if $t_{j+1} \rightarrow t_j$ in S_{UC} then

      t_j add to BcS_a and $a=\lambda(t_{j+1})$

      t_{j+1} add to BcS_a and $a=\lambda(t_j)$

for each a in $\mathcal{A}$ do

  $P_a^{Bcd}=|BcS_a|/|\mathcal{A}|, a=\lambda(t), P_a^{Bcd}$ add to S_p

Sort the elements in the S_p from big to small to get $S_p^{'}$

for each P_a^{Bcd} in $S_p^{'}$ do

  Remove the $\lambda^{-1}(a)$ from L to get $L^{'}$

  if $precision(L,M_L) > precision(L',M_{L'})$ and

  $|precision(L,M_L) - precision(L',M_{L'})| < 0.1$

//M_L and $M_{L'}$ represent the discovery models of L and L', respectively
  $L = L'$ continue
 else $L' = L$ break
return L'

4.2 Algorithm Sufficiency

Given a A-based event log L, $a \leftrightarrow_L b$ indicates bidirectional causal dependence between activity *a* and activity *b* where *a* and *b* belong to A.We can infer that in the process of model discovery, the transitions corresponding to activity *a* and activity *b* must be in parallel branches, and the path that the model can generate is 2: ab or ba. Similarly, we can know that if $a \leftrightarrow_L b, a \leftrightarrow_L c$, that is, there is a bidirectional causal dependence between activity *a* and activity *b* or activity *c*, then the possible path of the model is $3 \sim 3!$. If there is no relationship between activity *b* and activity *c* or there is a unidirectional causal dependence, the model may generate 3 paths, and if there is a bidirectional causal dependence between activity *b* and activity *c*, the model may generate 3! paths. When there is a bidirectional causal dependence between activity *a* and other activities, the corresponding model may generate $n \sim n!$ paths.

The emergence of bidirectional causal dependence leads to a large increase in the number of paths in the model, which causes a large number of traces of model not appear in the event log, that is the under-fitting of the model. This phenomenon is reflected in the model evaluation criteria is the precision between the event log and the model, so the algorithm proposed in this chapter finally uses precision as the criteria for activity selection.

4.3 Algorithm Efficiency Analysis

Let the event log contain n traces and m kinds of activities. The chaotic activity filtering approach based on information entropy first traverses the event log to obtain the relationship sequence of each activity relative to other activity lengths of *m+1*, and then traverses all the relationships. We traverse through all the relational sequences to calculate the entropy of each activity. In the computation of information entropy, the algorithm takes *2n(m+1)* to execute, that is, the time complexity of the algorithm is $O(n * m)$. The approach proposed in this paper traverses the event log to obtain a bidirectional causal dependence set for each activity relative to other activities. Assuming that the average length of the bidirectional causal dependency set is m', combined with the example in Table 2, we can find that $m' < m$. Finally, we can find that when there are many types of activities in the event log, the chaos activity filtering approach based on bidirectional causal dependence takes $n(m'+1)$ and $n(m'+1) \ll 2n(m+1)$.

From the previous bidirectional causal dependence set table, we can see that the probability of the bidirectional causal dependence P_X^{Bcd} corresponding to activity *X* is the largest, $P_X^{Bcd} = 0.444$, followed by the probability of bidirectional causal dependency of activity *D*, $P_D^{Bcd} = 0.333$, so we First delete the activity *X* from the event log and discover the new process model. Let the original event log and the model correspond to L' and $M_{L'}$ respectively. After deleting the activity *X*, the log and model correspond to *L* and M_L respectively. Through calculations, the $precision(L', M_{L'})$ between the original event log and the model equals 0.69798, and the $precision(L, M_L)$ between the log and the model after deleting the activity *X* equals 0.76641. According to the approach proposed in this paper, deleting the activity *D* in the event log *L* after deleting the activity *X*, we can get the log L'' as shown in Table 3, based on the log L'' through the Inductive Miner algorithm and finally finding the model $M_{L''}$ as shown in Fig. 3, then the accuracy between L'' and $M_{L''}$ is calculated to be 0.76658, further we can know $|precision(L'', M_{L''}) - precision(L, M_L)| \approx 0$. Therefore, we finally determine that activity *X* is filtered as chaotic activity in the original event log.

Table 3. Event Log L''

Event Sequences
$< A, B, E, H >$
$< A, C, E, G >$
$< A, C, E, F, B, E, G >$
$< A, B, E, H >$
$< A, C, E, F, C, E, F, C, E, H >$
$< A, C, E, G >$

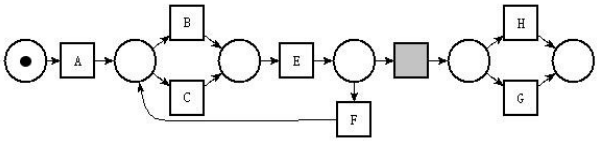

Fig. 3. Process Model Based on Event Data Log L''

5 Evaluation using Real Life Data

The data used in the experiment is the event log about the data customization process in the Tianyuan big data platform. The simulation experiment is completed on the ProM6.7 platform. By analyzing and experimenting with different magnitude event logs, we verify the correctness and effectiveness of the proposed approach.

5.1 Data and Model for Experiments

The process model used in this paper is the data customization process in the Tianyuan big data platform. Table 4 is the symbol and meaning matching table of the data customization process in the Tianyuan big data platform.

Table 4. Symbols and meanings

Symbols	Meanings	Symbols	Meanings
a	Create Account	*j*	Sign Contract
b	Understand Data Requirements	*k*	Data Collection
c	Data Needs Assessment	*l*	Data Cleaning
d	Communication via Email	*m*	Platform Data Release
e	Offline Communication	*n*	Customer Feedback
f	Communication via Platform	*o*	Offline Data Delivery
g	Demand Confirmation	*p*	Customer Confirmation
h	Make Data Template	*q*	Successful Delivery
i	Make Collection Plan	*r*	Customer Telephone Consultation

The entire data customization process begins with the user creating an account (*a*). When the account is created successfully, the user needs to understand the data requirements on the data platform (*b*), and if the user understands the data requirement in advance, this step can be skipped. The data needs assessment (*c*) is a link that must be performed before the user collects data, and the data needs assessment needs detailed communication between the user and the customer, and the communication manner includes email communication (*d*), offline communication (*e*) or directly through the data platform (*f*). The user needs to confirm the data requirement (*g*) after communicating with the customer and the user needs to make the data template (*h*) and collection plan (*i*) after confirming the requirement, and then sign the contract with the customer (j). The data collection (*k*) is the main link after the contract is signed, and the user needs to perform data cleaning (*l*) according to the requirements after the data collection is completed, and if the collected data meets the requirements, the data cleaning step can be skipped. When the data collection is completed, the user can deliver the data by publishing data (*m*) or offline (*o*), and the former must receive customer feedback (*n*). When the customer confirms (*p*) the data

delivery is successful (q). Customer consultation (r) can occur at any time after the user creates an account.

In the experiment process, we randomly selected five groups of event logs $L_1 \sim L_5$ for chaotic activity filtering, and the log cases included chaotic activities. Table 5 is the event log statistics table, in which each column in the table represents the total number of cases in the log, total number of events and the length range of traces.

Table 5. Event log statistics

Log	Total Number of Cases	Total Number of Events	Length of Traces
L_1	1023	15346	11-19
L_2	864	12589	13-16
L_3	1253	18268	11-17
L_4	1184	17994	12-20
L_5	967	14523	11-18

Taking event log L_1 as an example, the Inductive Miner algorithm performs model discovery. The resulting data customization process model is shown in Fig. 4.

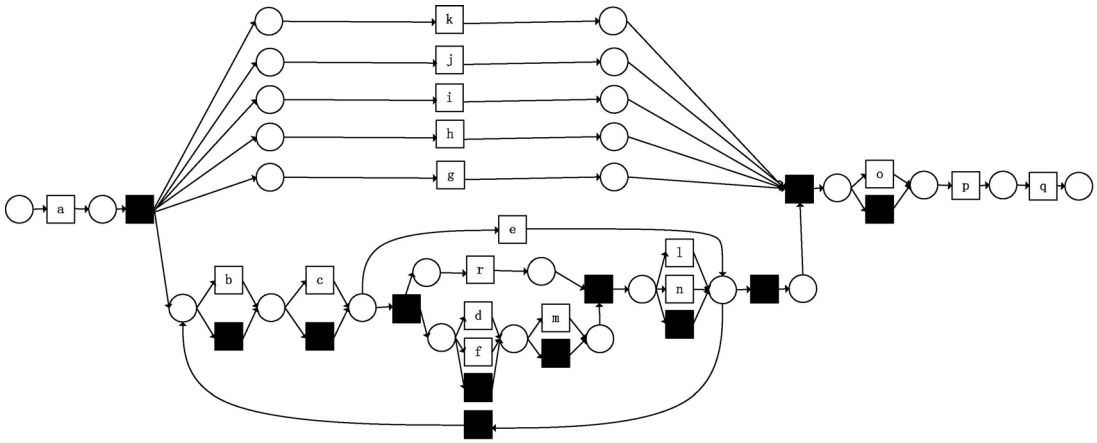

Fig. 4. Data customization process model based on event Log L_1 discovery

We can conclude from Fig. 4 that due to the existence of chaotic activities, there is a large deviation between the discovered model and the basic data customization process. Using the proposed approach to perform chaotic activity filtering on the event data log L_1 to get the event log L_1', finally get the data customization process model shown in Fig. 5. We can see from Fig. 5 that the model after chaotic filtering is more consistent with the basic data customization process. At the same time, it can be seen that there are more redundant paths in the model before chaotic activity filtering than in the model after chaotic activity filtering. The paper evaluates the model by measuring the precision between different models and the corresponding event logs to verify the effectiveness of the proposed approach.

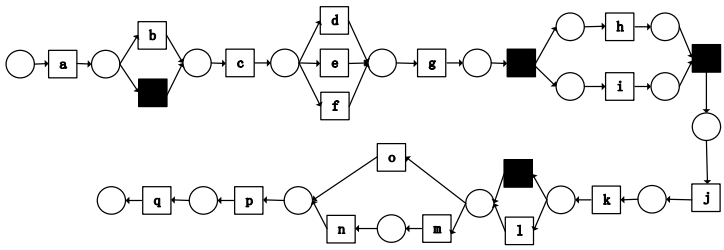

Fig. 5. Data customization process model based on event Log L_1' discovery

5.2 Contrastive Analysis of Different Techniques

In this section, we compare the precision of different event logs before and after chaotic activity filtering and the precision changes between the model and the log when the logs are deleted with different amounts of activity. Based on this, we evaluate the proposed approach.

For the event data log $L_1 \sim L_5$, we use the bidirectional causal dependence chaotic activity filtering approach to conduct experiments, and finally to the results shown in Fig. 6 by calculating the precision between the model and the corresponding log before and after filtering chaotic activities. It can be seen from Fig. 6 that the precision of the chaotic activity filtering is significantly higher than the value corresponding to the precision of the chaotic activity filtering.

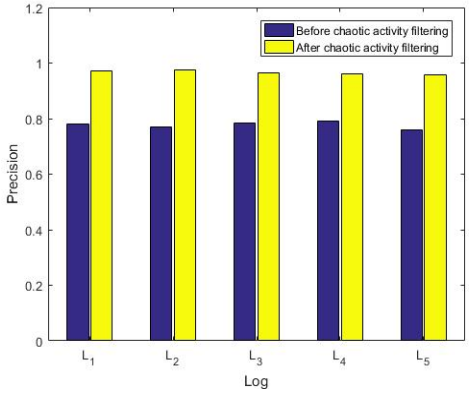

Fig. 6. Precision comparison between event logs and models before and after chaotic activity filtering

For the logs L_2 and L_3, we first calculate the probability of bidirectional causal dependence of different activities in the log, and then delete the different number of activities from the logs in the order of probability from high to low to get the corresponding model, and finally calculate the accuracy between the model and the log and get the results shown in Fig. 7. From Fig. 7 we can see that due to the elimination of chaotic activity, the precision between the model and the log is significantly improved, that is, the effect of chaotic activities on the precision of the model and event log is significantly higher than other activities.

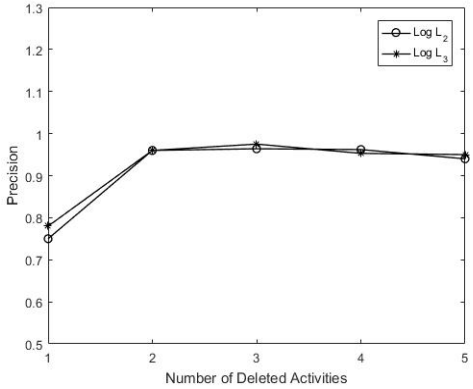

Fig. 7. Precision comparison between event logs and models when deleting different numbers of activities

For the logs $L_1 \sim L_5$, we use chaotic activity filtering approach based on bidirectional causal dependence and chaotic activity filtering approach based on information entropy to perform chaotic activity filtering, and calculate the time spent by different approach in filtering chaotic activities. The final result is shown in Fig. 8. From Fig. 8 we can see that the approach proposed in this paper is significantly less time-consuming than chaotic activity filtering based on information entropy.

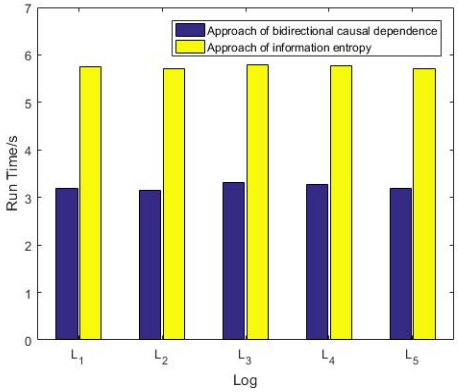

Fig. 8. Time of chaotic activity filtering

6 Conclusion & Future Work

This paper analyzes the influence of the existence of chaotic activity in the event log on the model discovery and finds that if there are chaotic activities in most traces of the event log, the use of existing filtering approach cannot solve the chaos activity in the event log. In order to solve the above problems, this paper proposes a chaotic activity filtering approach based on bidirectional causal dependence. By finding bidirectional causal dependencies between activities in the event log, it effectively filters the chaotic activity in the event log and obtains an accurate process model through process discovery.

Compared with the chaotic activity filtering approach based on information entropy, and calculating the precision of different magnitudes of event log filtering in the process of chaotic activity, the effectiveness and correctness of the proposed approach are verified.

Future work aims at extending and improving chaotic activities filtering in various ways.

Frist of all, we would like to analyze the log from the multiple perspectives in the filtering process of chaotic activities, such as time perspective or resource perspective.

Second, we are interested in considering supervised learning in data mining domain in the filtering process of chaotic activities.

Last but not least, one can consider the degree of simplicity and generalization of the process model in the evaluation of filtering chaotic activities.

Acknowledgements

This work is supported by the National Key Research and Development Plan No. 2016YFB1001100; the National Natural Science Foundation of China under grants No. 61272093; Petri Net innovation team of College of Computer Science and Engineering Shandong University of Science and Technology.

References

[1] W.M.P.V.D. Aalst, A.H.M.T. Hofstede, M. Wesk, Business process management: a survey, Lecture Notes in Computer Science 10(2)(2008) 1-12.

[2] M. Baklizky, M. Fantinato, L.H. Thom, V. Sum, P.C.K. Hung, Business process point analysis: survey experiments, Business Process Management Journal 23(2)(2017) 399-424.

[3] P. Harmon, The scope and evolution of business process management, in: J. vom Brocke, M. Rosemann (Eds.), Handbook on Business Process Management 1, Springer, Berlin, Heidelberg, 2015, pp. 37-80.

[4] B. Mutschler, M. Reichert, Understanding the costs of business process management technology, in: M. Glykas (Ed.), Business Process Management, Springer Berlin Heidelberg, 2013, pp. 157-194.

[5] W.M.P.V.D. Aalst, A.K.A.D. Medeiros, A.J.M.M. Weijters, Genetic Process Mining, Lecture Notes in Computer Science, 14(2)(2006) 76-83.

[6] W.M.P.V.D. Aalst, Process Mining: Data Science in Action, Springer, 2016.

[7] S.J.J. Leemans, Robust process mining with guarantees, [Ph. D. thesis] Urban, Dutch: Eindhoven University of Technology, 2017.

[8] A.A. Kalenkova, W.M.P.V.D. Aalst, I.A. Lomazova, V.A. Rubin, Process mining using BPMN: relating event logs and process models, in: Proc. the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, 2016.

[9] J.C.A.M. Buijs, B.F.V Dongen, W.M.P.V.D. Aalst, A genetic algorithm for discovering process trees, in: Proc. 2012 IEEE Congress on Evolutionary Computation (CEC), 2012.

[10] S. Goedertier, D. Martens, J. Vanthienen, B. Baesens, Robust process discovery with artificial negative events, Journal of Machine Learning Research, 10(Jun)(2009) 1305-1340.

[11] C. Günther, W.M.P.V.D. Aalst, Fuzzy mining–adaptive process simplification based on multi-perspective metrics, in: Proc. Business Process Management, 2007.

[12] J. Herbst, A machine learning approach to workflow management, in: Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2000.

[13] M. Solé, J. Carmona, Region-based foldings in process discovery, IEEE Transactions on Knowledge and Data Engineering 25(1)(2013) 192-205.

[14] S.J.J. Leemans, D. Fahland, W.M.P.V.D. Aalst, Discovering block-structured process models from event logs containing infrequent behaviour, in: Proc. International Conference on Business Process Management. Springer, 2013.

[15] S. Suriadi, R. Andrews, A.H.M.T. Hofstede, M.T. Wynn, Event log imperfection patterns for process mining: towards a systematic approach to cleaning event logs, Information Systems, 64(2017) 132-150.

[16] B.F.V. Dongen, A.K.A.D. Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, W.M.P. van der Aalst, The prom framework: A new era in process mining tool support, in: Proc. International Conference on Applications and Theory of Petri Nets and Concurrency, 2005.

[17] R. Conforti, M.L. Rosa, A.H.M.T. Hofstede, Filtering out infrequent behavior from business process event logs, IEEE Transactions on Knowledge and Data Engineering 29(2)(2017) 300-314.

[18] L. Ghionna, G. Greco, A. Guzzo, L. Pontieri, Outlier detection techniques for process mining applications, Lecture Notes in Computer Science 4994(2008) 150-159.

[19] N. Tax, N. Sidorova, W.M.P.V.D. Aalst, Discovering more precise process models from event logs by filtering out chaotic activities, arXiv preprint arXiv:1711.01287, 2017.

[20] A. Rozinat, W.M.P.V.D. Aalst, Conformance checking of processes based on monitoring real behavior, Information Systems 33(1)(2008) 64-95.

[21] F. Mannhardt, M. de Leoni, H.A. Reijers, W.M.P. van der Aalst, Balanced multi-perspective checking of process conformance, Computing 98(4)(2016) 407-437.

[22] J.M. Gama, J. Carmona, W.M.P.V.D. Aalst, Conformance checking in the large: partitioning and topology, in: Proc. International Conference on Business Process Management, 2013.