An efficient process mining model for Petri Nets in process discovery

V.Priyadharshini[1], Dr. A. Malathi[2]

Ph.D. Research Scholar [1], Assistant Professor[2]

[1,2]PG & Research Department of CS Government Arts College (Autonomous) Coimbatore – 18

Abstract: *Process mining is a process management system used to analyze business processes based on event logs. The knowledge is extracted from event logs by using knowledge retrieval techniques. The process mining algorithms are accomplished of inevitably discover models to give details of all the events registered in some log traces provided as input. The theory of regions is a valuable tool in process discovery: it aims at learning a formal model (Petri nets) from a set of traces. The main objective of this paper is to propose new concepts for noise filtering and label splitting problem. The experiment is done based on standard bench mark dataset HELIX and RALIC datasets. The performance of the proposed system is better than other existing methods.*
Keywords: *Process Discovery, Process Mining, Noise filtering, Label splitting, HELIX and RALIC Datasets.*

I. Introduction

Process discovery is one of the most challenging process mining tasks. Based on event log, a process model is constructed by capturing the behavior of the log [1]. Event logs essentially capture the business activities happened at a certain time period [2]. The basic plan is to extract data from event logs recorded by a data system. The method mining aims at rising this by providing techniques and tools for locating method, control, data, structure, and social structures from event logs. The research domain that is concerned with knowledge discovery from event logs is called process mining [3]. More traditional data mining techniques can be used in process mining. New techniques are developed to perform process mining i.e. mining of process models. It is the traditional analysis of business processes based on the opinion of process expert [4]. The business process mining attempts to reconstruct a complete process model from data logs that contain real process execution data [5]. Many techniques highlight the possibility of combining a number of process mining approaches to mine more interesting event logs, such as those that contain noise.

The necessary background in Section II describes related work. Section III presents existing algorithms that describes previous work done. Section IV describes the proposed implemented work. The result and discussion is presented in Section V. Conclusion and future work is discussed in Section VI.

II. Related Survey

Fabio in [6] compared business process models with metric that is based on a representation of local relations between pairs of activities in a model. Saravanan in [7] developed a control flow process mining algorithm that can discover all the common control flow structures. Boleslaw in [8] described a workflow mining algorithm that is used to improve and/or refine design of existing workflows. The time and space complexity is dependent on workflow's internal structure and on the number of workflow tasks.

Process mining aims at inferring process and knowledge from event logs thus providing diagnosis tools. Process discovery techniques aim at reconstructing business process model from event logs [9]. A genetic process discovery algorithm has certain quality criteria that cannot guarantee for noise filtering in finite run time [10]. The language based mining technique uses regular expressions to pre structure the input log traces into smaller blocks [11]. The classical synthesis technique based on regions cannot be applied directly because the event log contains only example behavior [12].

III. Existing System

1.1 Alpha algorithm

The **α-algorithm** is an algorithm used in process mining, aimed at reconstructing causality from a set of sequences of events. It constructs P/T nets with special properties (workflow nets) from event logs [13]. Each transition in the net corresponds to an observed task. It orders events sequentially, such that each event refers to a case and activity. It has problems with noise, infrequent behavior and complex routing constructs [14]. It scans event logs for particular patterns. It makes single pass through workflow log to identify which tasks directly follow each other. It is sensitive to noise and incompleteness of event logs [15]. The algorithm takes a workflow log as input and results in a workflow net being constructed. It will thus by examining causative relationships

determined between tasks [16]. For instance, one specific task would possibly continually precede another specific task in each execution trace, which might be helpful data.

1.2 Cycle Detection Algorithm

The region information in the cycle detection algorithm is incorporated by enabling the detection of complex cycles that cannot be obtained efficiently with state-of-the-art techniques [17]. This approach is discovering Petri net in an efficient way. But this method does not reject a trace of the log; they are very sensitive to noise. A low support threshold that is high enough to reduce the number of Frequentgroup patterns to a manageable level is used [18]. The intermediate step in the region-based Process Discovery flow; a folding step that is performed before applying a process discovery method, which allows factoring out the cycles and therefore reduces the size of the transition system [19].More significantly, dissimilarity with approaches that may yield transition structure with cycles, this decrease in size comes with the guarantee that significant subsets of regions are preserved [20].

It is a pointer algorithmic rule that uses solely two pointers that move through the sequence at totally different speeds. The algorithmic rule solely must check for recurrent values of this special type, one double as aloof from the beginning of the sequence because the different, to search out an amount. The function value is used to find a cycle in a sequence of iterations. Cycles are available in a graph and in much real life application; it is required to know the existence of cycles in a graph [21]. The algorithm is developed in the context of network design problem but useful in any graph application where existence is to be finding out.

IV. Proposed System

In this work we propose new approaches for noise filtering and Label splitting problem. The main objective of this paper is to propose a conversion for k-bounded discovery that produces more compact TSs than the traditional conversion methods, without losing any precision [22]. Although this technique can be applied to any TS regardless of the conversion used, here develop scheme on top of the sequence and multiset changes, since the other changes are not guaranteed to reserve the same set of regions as the other two do [23]. For region-based discovery algorithms yielding a k bounded PN, this state equivalence description can be distinguished, since the algorithm will only reflect the k bounded regions of the TS. This approach is discovering Petri net in an efficient way. But this method does not reject a trace of the log; as they are very sensitive to noise [24].

4.1 Unsupervised Noise filtering Algorithm

The proposed system automatically decides correct initial weight, noise filtering, and feature selection properties. This proposed algorithm not only does noise filtering, it also measures the relationship of members in all calculated clusters to find the clarity of relationship. If any record has uncertain membership, it will be considered as the noise.

4.2 Frequentgroup based noise filtering Algorithm

It finds all Frequentgroup patterns for a given minimum support and minimum confidence and remove the objects that are not a part of any Frequentgroup pattern. The set of Frequentgroup patterns for any data set depends upon the value of minimum support and minimum confidence. Wherever possible, we fix the minimum support to be zero and employ the minimum confidence to control the number of objects that are designated as noise. However, in some data sets setting the support threshold to zero leads to an explosion in the number of Frequentgroup patterns. For this reason, a low support threshold that is high enough to reduce the number of Frequentgroup patterns to a manageable level is used. The proposed system is more efficient than the existing system.

4.3 Label Splitting Problem

This work is applying label splitting a technique that is used to improve the visualization of the PN; it is also used to avoid over generalization. In some cases the derived Petri net equivalent to the initial transition system is required. This cannot be always accomplished. This problem is solved by defining a property excitation closure that TS must satisfy in order to derive a PN with bi-similar behavior. A label splitting technique is presented here is used to repair excitation closure violations in a Transition system. This allows to present a method for obtain bi-similar k-bounded PNs; excitation closure is used to reduce the number of necessary regions while preserving the properties in the derived net.

Two open problems are formulated in the previous work,
- If the existence of an optimal net can be characterized in terms of the TSs and
- If there always exists at most one optimal net which could be considered canonical.

It gives a positive answer to the first problem and a negative answer to the second problem.

4.4 Datasets

Helix and RALIC datasets is a compilation of release histories of a number of non-trivial Java Open Source Software System. It contains class files for each release of the system along with meta-data. A metric history is derived from extraction of releases and this data is directly used in research works.

V.   Result And Discussion

In the work helix and RALIC dataset are used for the experiment. We are comparing the three algorithms for noise filtering and label splitting problem. In other words, the proposed unsupervised noise filtering and Frequentgroup based noise filtering and Label splitting. The fitness value of noise filtering is low compared to the proposed noise filtering and label splitting techniques. By using proposed label splitting in this system, we obtain the high fitness value. So we can obtain the effective performance comparatively.
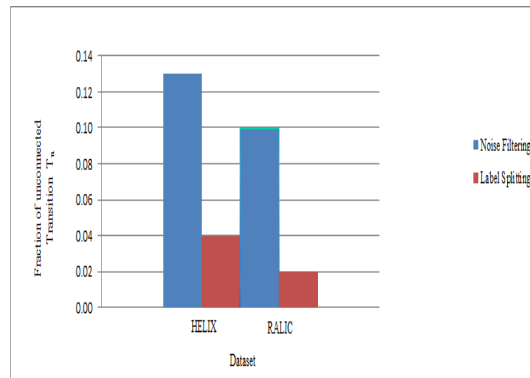

Fig 1. Fitness value for fraction of unconnected Transitions

The experiment is based on fitness value calculation and fraction of unconnected transitions (T_u). The second experiment based on fraction of unconnected transitions (T_u), we calculate the T_u corresponding to existing and proposed approaches. The T_u is decreased for the proposed methods label splitting technique. This will also increase the performance of the proposed system compared to existing system. Finally we can conclude as the proposed label splitting approach has more effective than the existing system as shown in Fig 1.

In Fig 2. Three proposed algorithms are compared. They are noise filtering and label splitting approaches. In this comparison, the parameter for the comparison is time complexity. The time complexity is defined as the time taken for performs the operation of the technique. The proposed label splitting has lowest complexity of time compared to the other two techniques. Since, this proposed system reduces the time complexity it is very effective.
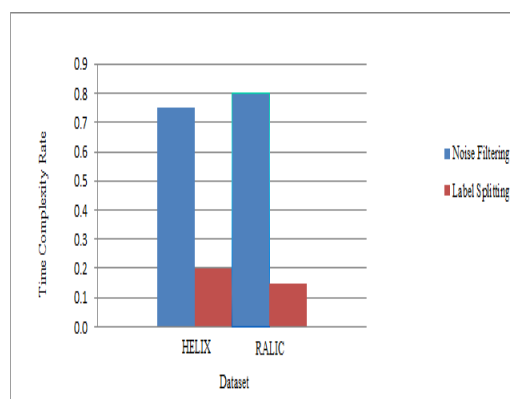

Fig 2. Time Complexity Rate

VI.   Conclusion And Future Work

The datasets were used with three different region based algorithms for unconnected transitions and the result is shown. In future, we will look for improvements of the existing process discovery and visualization techniques that allow for the construction of comprehensible models based on realistic characteristics of an event logs.

References

[1] Aalst, W.M.P.v.d., et al., Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16(9) (2004) 1128–1142.

[2] S.J.J.Leemans et al, Discovering block-structured process models from event logs- A constructive approach.

[3] Joonsoo Bae et al, Development of distance measures for process mining, discovery and integration, International journal of web services research, Vol.10, No.10, 2010.

[4] Process Mining Manifesto. IEEE Task Force on Process Mining.

[5] Arya Adriansyah, Peter van den brand et al, Process mining manifesto, Business process management, Volume 99, 2012, pp. 169-194.

[6] Fabio Aiolli, Andrea Burattin, and Alessandro Sperduti. A Business Process Metric Based on the Alpha Algorithm Relations.

[7] Saravanan .M.S, Rama Sree .R.J. A Role of Heuristics Miner Algorithm in the Business Process System, International Journal of Computer Technology and Applications, 340-344.

[8] Boleslaw Mikolajczak, Jian-Lun Chen. 2005. Workflow Mining Alpha Algorithm — A Complexity Study, Intelligent Information Processing and Web Mining Advances in Soft Computing, 451-455.

[9] Aalst, W.M.P.v.d., Gunther C.W., et al. 2010. Process mining: A two-step approach to balance between under fitting and over fitting. Software and Systems Modeling, 87–111.

[10] De Medeiros, A., Weijters, A., and van der Aalst, W. 2007. Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery, 245–304.

[11] Filip Caron, Jan Vanthienen and Bart Baesens. A comprehensive framework for the application of process mining in risk management and compliance checking, Faculty of Business and Economics.

[12] Jochen De Weerdt et al. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs.

[13] C.J. Turner, A. Tiwari, R. A. Olaiya and Y, Xu, Business Process Mining: From Theory to Practice, Business Process Management Journal, 2012, Vol. 18 Issue: 3, pp.493 – 512.

[14] Josep Carmona et al, New region-based algorithms for deriving bounded petri nets, IEEE Transactions on Computers, Vol. 59, No.3, March 2010.

[15] Gianluigi Greco et al, Discovering expressive process models by clustering log traces, IEEE Transactions on Knowledge and data engineering, Vol. 18, No. 8, Aug 2006.

[16] Bergenthum, R., Desel, J., Mauser, S., Lorenz, R., Synthesis of Petri nets from term based representations of infinite partial languages. Fundamentals in Information System. 95(1), 187–217 (2009).

[17] W.M.P. Van der Aalst, The application of Petri nets to work flow management.

[18] Gianluigi Greco et al, Discovering expressive process models by clustering log traces, IEEE Transactions on Knowledge and data engineering, Vol. 18, No. 8, Aug 2006.

[19] Jochen De Weerdt et al, Active trace clustering for improved process discovery, IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 12, Dec 2013.

[20] Van der Aalst, W., de Medeiros, A., Weijters, A., Genetic process mining. Applications and Theory of Petri Nets 2005 pp. 985–985 (2005).

[21] Jordi Cortadella et al, Deriving Petri Nets from Finite Transition Systems, IEEE Transactions on Computers, Vol. 47, No.8, Aug 1998.

[22] Bergenthum, R., Desel, J., Mauser, S., Lorenz, R., Synthesis of Petri nets from term based representations of infinite partial languages. Fundamentals in Information System. 95(1), 187–217 (2009).

[23] Josep Carmona et al, New region-based algorithms for deriving bounded petri nets, IEEE Transactions on Computers, Vol. 59, No.3, March 2010.

[24] Zhiang Wu et al, A novel noise filter based on interesting pattern mining for bag-of-features images, Expert systems with applications 2013.