UNIVERSITÄT KOBLENZ · LANDAU

Fachbereich 4: Informatik

Probabilistic Social Process Mining

Masterarbeit

zur Erlangung des Grades eines Master of Science
im Studiengang Web Science

vorgelegt von

Rahul Arora
Mat.-Nr. 213202976
arora@uni-koblenz.de

Erstgutachter: Patrick Delfmann
Institut für Wirtschafts- und die Verwaltungsinformatik

Zweitgutachter: Carl Corea
Institut für Wirtschafts- und die Verwaltungsinformatik

Koblenz, im Oktober 2017

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

. . . . . .

(Ort, Datum) (Unterschrift)

Abstract

This thesis explores the possibilities of probabilistic process modelling for the Computer Supported Cooperative Work (CSCW) systems in order to predict the behaviour of the users present in the CSCW system. Toward this objective applicability, advantages, limitations and challenges of probabilistic modelling are excavated in context of CSCW systems. Finally, as a primary goal seven models are created and examined to show the feasibilities of probabilistic process discovery and predictions of the users behaviour in CSCW systems.

Acknowledgement

I'd like to thank my supervisor Prof. Dr. Patrick Delfmann and his PHD student Christoph Drodt for their support throughout this thesis.

I'd like to thank my parents for believing in me and supporting me throughout all my studies.

Contents

1 Introduction **12**

1.1 Context of this work 13

1.2 Related work 14

1.3 Research goals 16

1.4 Research methodology 17

2 Research foundations & glossary **19**

2.1 Introduction to dataset 19

2.2 Naming convention & tool used 21

2.3 Process mining 26

3 RegPFA and RapidProm **30**

3.1 RegPFA 30

3.1.1 Probabilistic models 30

3.1.2 Component first: RegPFA predictor 31

3.1.3 Model scorers (KPIS) 34

3.1.4 RegPFA parameters 35

3.1.5 Component second: RegPFA analyzer 36

3.2 Rapid Miner and RapidProm 37

3.2.1 RegPFA operator 39

4 Outcomes and evaluation of created models 40

4.1 Model 0: naive model 41

4.2 Models methodology 42

4.3 Model one: predicting users activity 45

4.3.1 Introduction to the case 45

4.3.2 Log filtering 46

4.3.3 RegPFA mining & parameter modification 49

4.3.4 User behaviour prediction 54

4.4 Model two: predictive modelling impact on mainstream users 58

4.4.1 Introduction to the case 58

4.4.2 Log filtering 58

4.4.3 RegPFA mining & parameter modification 60

4.4.4 Mainstream user behaviour prediction 63

4.5 Model three: prediction of communities that users will join. 65

4.5.1 Introduction to the case 65

4.5.2 Log filtering 66

4.5.3 RegPFA mining & parameter modification 67

4.5.4 Community joining predictions 70

4.6 Model four: predictive modelling for mainstream user behaviour 73

4.6.1 Introduction to the case 73

4.6.2 Log filtering 73

4.6.3 RegPFA mining & parameter modification 75

4.6.4 User behaviour prediction 76

4.7 Model five: predicting activities performed by users in communities . . . 77

4.7.1 Introduction to the case 77

4.7.2 Log filtering 78

4.7.3 RegPFA mining & parameter modification 78

4.7.4 User behaviour prediction 81

4.8 Model six: predicting events that will happen inside communities 84

4.8.1 Introduction to the case 84
4.8.2 Log filtering 85
4.8.3 RegPFA mining & parameter modification 86
4.8.4 Activities prediction 88
4.9 Models findings 91

5 Conclusion **95**
5.1 Summary 95
5.2 Future Work 97

List of Figures

1.1 Two Step approach for process discovery and user prediction (source:self-made) 14
1.2 Structure of event logs and CSCW logs (source:(Günther & Rozinat 2012, van der Aalst 2007). 16

2.1 Step by step marking for conversion of CSV to XES in Disco (source:self-made, inspired by (Günther & Rozinat 2012)). 22
2.2 Process map (flow) visualization In Disco (source:self-made). 23
2.3 Statistics about process instance and individual cases (source:self-made). . 24
2.4 Variants containing different cases and events (source:self-made). 25
2.5 Filters available in Disco (source:self-made). 25
2.6 Types of process mining source: (Van 2011) 28

3.1 Overview of the RegPFA components (source:Breuker et al. (2016)). 32
3.2 Workflow in RapidMiner (source:self-made). 38

4.1 Methodology used for models (source:self-made) 42
4.2 Rapid miner process workflow used for all the models (source:self-made). 43
4.3 Empty activities indicated by a black rectangle (source:self-made). 45
4.4 Cases with few events indicated by a black rectangle (source:self-made). . 46
4.5 KPIs for the best fitted process model (source:self-made). 53
4.6 HIC for the best fitted process model (source:self-made). 54

4.7 Camouflaged visualization for model one (source:self-made). 55
4.8 Unobservable process for model one (source:self-made). 56
4.9 Best fitted process model with 0.0004 pruning ratio for model one (source:self-made) 57
4.10 Exceptional behaviour in logs (source:self-made) 58
4.11 Mainstream behaviour in logs (source:self-made) 59
4.12 HIC for the best fitted process model of model 2 (source:self-made). . . . 62
4.13 Latent visualizations for model two (source:self-made). 64
4.14 Optimal process model with pruning ratio 0.08 for model two (source:self-made). 65
4.15 High cumulative frequency of few activities indicated by a black rectangle (source:self-made). 66
4.16 HIC plot for the best fitted process model of model three (source: Self-made). 70
4.17 camouflaged visualization for model three (source:self-made). 71
4.18 Best fitted process model with Pruning ratio 0.1 for model 3 (source:self-made). 72
4.19 Disco showing cases that share common sequence of activities. 74
4.20 HIC for the best fitted process model of model four (source:self-made). . 76
4.21 Singleton activities shown by Disco (source:self-made). 77
4.22 HIC plot for model five (source:self-made). 80
4.23 Transition state diagram with pruning ratio 0.03 for model five (source:self-made). 82
4.24 Transition state diagram with pruning ratio 0.015 for model five (source:self-made). 83
4.25 Best fitted process model with pruning ratio 0.01 for model 5 (source:self-made). 84
4.26 Disco showing high relative frequency of empty activities (source:self-made). 85

4.27 HIC plot for best fitted model of model six. 88
4.28 Unobservable process model of model six (source:self-made). 89
4.29 Transition state diagram with 0.04 pruning ratio (source:self-made). . . . 89
4.30 Comprehensive process model for model 6 with 0.0004 pruning ratio (source:self-made). 90
4.31 Predictions for model six (source:self-made). 91

List of Tables

4.1 Parameters for naive run. 41
4.2 Performance filter applied first that is filtering events in the first place. . . 47
4.3 Attribute filter applied first that is filtering activities in the first place. . . . 47
4.4 Statistics after the performance filter i.e., to normalize case duration. . . . 48
4.5 Parameter details for the first iteration of model one. 49
4.6 Comparison of the two best models with EM iterations 100 and 500. . . . 50
4.7 Parameter details for the third iteration of model one. 51
4.8 Parameter details for the best fitted process model. 51
4.9 Parameter details for the best fitted process model of model one. 52
4.10 Statistics after the variation filter. 59
4.11 Parameter details for the first iteration of model two. 60
4.12 Parameter details for the second iteration of model two. 62
4.13 Statistics before and after applying the attribute filter in model three. . . . 67
4.14 Parameter details for the first iteration of model three. 68
4.15 Parameter details for the second iteration of model three. 68
4.16 Parameter details for the third iteration of model three. 69
4.17 Statistics before and after applying the attribute filter in model four. . . . 74
4.18 Parameter details for the first iteration of fourth model. 75
4.19 Statistics before and after applying the attribute filter in model five. 78
4.20 Parameter details for the first iteration of model five. 79
4.21 Parameter details for the second iteration of model five. 80

4.22 Statistics before and after applying the attribute filter in model six. 86
4.23 RegPFA parameter for the first iteration of model six. 87
4.24 Parameter details for the second iteration of model six. 87
4.25 Parameter details for the number of events between 50000 and 75000 . . 92
4.26 Parameter details for the number of events around 100000. 92
4.27 Parameter details for the number of events between 50000 and 75000. . . 93

Chapter 1

Introduction

Process aware information systems (PAISs) such as Enterprise Resource Planning (ERP), WorkFlow Management (WFM) and Business Process Management (BPM) have structured processes that possess a contrasting amounts of hidden information in the form of event logs (Ma 2007). This non-trivial information can be extracted in form of visual models through process mining techniques (Van der Aalst et al. 2004, Cook & Wolf 1998, Weijters & Van der Aalst 2003). However, van der Aalst (2007), Syri (1997) prove that process mining is not only limited to PAISs that have more structured processes, but can be applied to Computer Supported Cooperative Work (CSCW) systems as well, which contain the less structured processes. Process mining is gaining popularity in many areas and CSCW is no different (Van Der Aalst et al. 2005).

Process mining techniques are proven to be useful in improvement and better understanding of business processes, by unlocking the causal dependencies in event logs (van der Aalst et al. 2007) but does not provide any real time information about the underlying process. But the need of predictive analytics is surging in information systems (Shmueli & Koppius 2011) and some recent work in CSCW systems also reckoned its feasibility by predicting user behaviour (Yu et al. 2017). Therefore, to attain comprehensive insight about present and future events of the business processes in an operational setting, some pioneers of process mining field incorporated different approaches of predictive analytics

to it (Van der Aalst et al. 2011, Dongen et al. 2008, Schonenberg et al. 2008). One such approach was developed by Breuker et al. (2016) using the Probabilistic Finite Automaton (PFA) model in grammatical inference, section 3.1 discusses this approach in further detail. This approach not only predicts the behaviour of currently running business processes but also provides a comprehensive visualization so that users with less technical knowledge can have insight into the produced model. Moreover, this approach serves as the basis for this thesis by assisting in exploring the possibilities of probabilistic process mining in CSCW systems. Furthermore, it also supports in investigating the different perspectives of user behaviour predictions in CSCW systems.

1.1 Context of this work

This thesis discusses the opportunities of predictive process discovery for a particular CSCW system called UniConnect. UniConnect is a collaboration platform by University Competence Center for Collaboration Technologies (UCT), section 2.1 discusses UCT and UniConnect in more details. Whereas, the predictive approach (RegPFA (section 3.1)) which is used for this work is based on grammatical inference and is developed by Breuker et al. (2016). It states ”grammatical inference techniques can be applied to event logs to create a model that describes the corresponding business process, as the central learning problems of both fields are similar”. More specifically, it uses the PFA from grammatical inference because in business processes state variables depend on the preceding state variables and on the events observed during process execution. Figure 1.1 depicts an overview of the approach used for this thesis. The general idea of the approach is to use CSCW logs files, in particular like UniConnect, to mine process models out of the logs in a first step. In the second step these process models (probabilistic transition system (section 3.1)) are used for user behaviour prediction. As CSCW systems are highly collaborative, predictions describes the activities of business processes that users are highly likely to perform in the future. In this context this work mine seven different models for process discovery by determining the optimum parameter settings

for the RegPFA algorithm in Rapid miner (section 3.1 & 3.2) in order to provide various predictions about users activities.

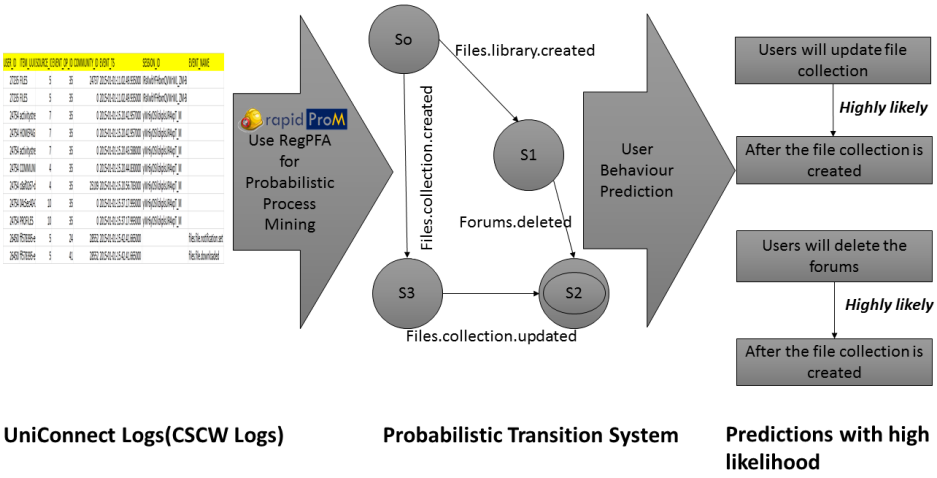


Figure 1.1: Two Step approach for process discovery and user prediction (source:self-made)

The next section discusses the related work and formulates the motives of this thesis while specifying the needs for the selection of RegPFA as an algorithm to fulfill the research goals of this work.

1.2 Related work

Initial work on process discovery was started in the late-nineties when Cook & Wolf (1998) discover the process models for the software engineering processes. The idea itself of mining the process models from the event logs of workflow management system was introduced by Agrawal et al. (1998). However, these approaches overlook the processes with duplicate tasks i.e., same task can appear multiple times in the workflow model. To overcome this Herbst (2000*b*,*a*) developed an approach that accounts for the duplicate task as well as concurrency issue, but the graph generation technique (process model) was similar to the traditional approaches, where transformation of the stochastic graph into a

particular workflow model was done. This approach captures the issue of parallelism in worklow processes, but by adding a specific mechanism. The general idea of concurrency in workflow mining was introduced by Van der Aalst et al. (2003, 2004) in the form of α-algorithm. However, this algorithm had problems dealing with noise and incompleteness. Heuristic miner and fuzzy miner were two robust technique that solve the problem of disorderliness and missing values in event logs (Weijters & Van der Aalst 2003).

The approaches described above discover process models from events logs but failed to provide real time information to users. Providing recommendations to users for selecting the next item based on historic information was one of the elementary step in delivering real time information (Schonenberg et al. 2008). Prediction of time for the case completion based on linear regression, formal and logic models were also presented (Dongen et al. 2008, Van der Aalst et al. 2011). These approaches named as two-step process discovery algorithm as it discover the process model and annotate it with the real-time data. These algorithms considered the events either as a sequence or a set to give the process a unique state at any point of time in order to built a corresponding automaton or grammar. These bases posture the possible predictions about the process instances such as, time remaining for a case to be completed. However, apart from the formal and logic models approaches in predictive process mining, there have been some traditional probabilistic approaches since the evolution of process mining (Datta 1998, Herbst & Karagiannis 2004). Jeong et al. (2010) used the HMM (Hidden Markov model) to mine the process model analyzing students learning behavior and Weber et al. (2013) applied PFA to α-algorithm. Together with traditional approaches, both of the aforementioned approaches prove that probabilistic techniques can handle the noisy data while providing the real time information about the process instances. However, there were some disadvantages of these probabilistic approaches:

- Graph (such as petri net) building technique of these approaches is similar to formal models i.e. first discover the process model and then annotate it with probabilistic information, as probabilistic techniques do not deliver a visual and executable model directly.

- They can lead to the problem of overfitting

To overcome these disadvantages in the area of predictive process mining Breuker et al. (2016) design a probabilistic technique called RegPFA for the event data, which is based on PFA and model a probability distribution directly over the set of all conceivable event sequences instead of first discovering a model structure and subsequently annotating probabilistic information. This approach undertook the issue of overfitting (log incompleteness) and comprehensibility by using Bayesian regularization and providing the transition state diagram directly as a result (section 3.1) respectively. Moreover, this approach suffice the basis of this thesis by applying it to the CSCW system since process mining is getting prevalent in CSCW systems (van der Aalst 2007) due to their similar log structure with event logs which is shown in figure 1.2a and 1.2b. Besides that some recent work shows the surging need of user behaviour prediction in CSCW system due to their highly collaborative nature (Yu et al. 2017). Hence, this work discusses the predictive process modelling (RegPFA, in particular) in context of CSCW domain.

(a) EventLogs

Fragment of Staffware log

```
Case 21
Diractive Description    Event            User                yyyy/mm/dd hh:mm
-------------------------------------------------------------------------------
                         Start            swdemo@staffw_edl 2003/02/05 15:00
Register order           Processed To     swdemo@staffw_edl 2003/02/05 15:00
Register order           Released By      swdemo@staffw_edl 2003/02/05 15:00
Prepare shipment         Processed To     swdemo@staffw_edl 2003/02/05 15:00
(Re)send bill            Processed To     swdemo@staffw_edl 2003/02/05 15:00
(Re)send bill            Released By      swdemo@staffw_edl 2003/02/05 15:01
Receive payment          Processed To     swdemo@staffw_edl 2003/02/05 15:01
Prepare shipment         Released By      swdemo@staffw_edl 2003/02/05 15:01
Ship goods               Processed To     swdemo@staffw_edl 2003/02/05 15:01
Ship goods               Released By      swdemo@staffw_edl 2003/02/05 15:02
Receive payment          Released By      swdemo@staffw_edl 2003/02/05 15:02
Archive order            Processed To     swdemo@staffw_edl 2003/02/05 15:02
Archive order            Released By      swdemo@staffw_edl 2003/02/05 15:02
                         Terminated                         2003/02/05 15:02

Case 22
Diractive Description    Event            User                yyyy/mm/dd hh:mm
-------------------------------------------------------------------------------
                         Start            swdemo@staffw_edl 2003/02/05 15:02
Register order           Processed To     swdemo@staffw_edl 2003/02/05 15:02
Register order           Released By      swdemo@staffw_edl 2003/02/05 15:02
Prepare shipment         Processed To     swdemo@staffw_edl 2003/02/05 15:02
```

(b) CSCWLogs.

Figure 1.2: Structure of event logs and CSCW logs (source:(Günther & Rozinat 2012, van der Aalst 2007).

1.3 Research goals

There are two main goals of this research which are as follows:

- *Exploring the feasibilities of predictive process mining in CSCW systems:* Process mining is a novel topic of research in the area of CSCW, and only few approaches have been designed until now (van der Aalst 2007, Van Der Aalst et al. 2005). However, this research depart from established theory and pursue a different approach that is based on predictive process modelling and builds a relationship between process mining techniques and probabilistic grammatical inference techniques (Breuker et al. 2016). Additionally, it proved effective and can compete with state-of-the-art techniques from process mining. Therefore, we explore the opportunities that arise from applying probabilistic grammatical-inference techniques to CSCW.

- *User behaviour Prediction in CSCW system:* Due to the highly collaborative nature of CSCW systems users perform different activities collaboratively and coordinately in them (Carstensen & Schmidt 1999). In light of this it would be interesting to design the CSCW processes accordingly if designers know in advance which series of activities are highly likely that a user will perform in near future as such understanding helps them in improving the system performance. Therefore, to predict the user behaviour accurately in CSCW systems predictive process mining can be adapted as technique by considering the user related fields as the inputs of the corresponding technique. The technique will then provide a probabilistic process model in terms of activity performed by the user, which in turn can be used to predict user behaviour.

1.4 Research methodology

This research follows the behavioral research paradigm by conducting an exploratory research. According to Shields & Rangarajan (2013) ”exploratory study is conducted for a problem that has not been studied more clearly, establishes priorities, develops operational definitions and have few studies to rely upon to predict an outcome”. The main goals of exploratory study is to get familiar with basic details and environment settings in order

to build a well grounded picture of the situation being developed Mills et al. (2010). In the line of argumentation provided by Shields & Rangarajan (2013), Mills et al. (2010), this thesis explore the possibilities of probabilistic process modelling in CSCW system in order to predict the user behaviour from the corresponding probabilistic model and is organized as follows:

Chapter two lays the foundations of research by providing an introduction to the dataset, Disco naming conventions that will be used through out this thesis and finally calls the need of probabilistic process model while describing the process mining terminology (Van Der Aalst & Adriansyah 2011).

Chapter three discuss the general setting by describing the grammatical inference and its two most important probabilistic models i.e., HMM and PFA (De La Higuera 2005, Verwer et al. 2014). It then introduce the methodological apparatus used for this research in terms of RegPFA and Rapid miner (Breuker et al. 2016), by notifying the need of various models for this thesis.

Initially, chapter four explained the methodology that is used for all the models. Then all the models are discussed in greater detail, followed by evaluation of all the models according to their respective knowledge performance indicator (KPIs). The best model is then used to predict the user behaviour. Finally this chapter ends with the findings of all the models which can used when applying the RegPFA in other CSCW system.

Final chapter discusses the summary of this work by outlining some directions for future work.

Chapter 2

Research foundations & glossary

This chapter describes the dataset in detail by stating how event log conventions can be adopted to CSCW systems. Moreover, it lays the foundations of basic concepts and terminology of process mining. Furthermore, it describe the type and evaluation criteria of process mining that can be used for probabilistic process mining (RegPFA) to full-fill the research goals of this work .

2.1 Introduction to dataset

The input for the process mining step are event logs from productive operational systems like SAP or workflow engines. However, the goal of this research is to explore the possibilities of predictive process modelling in CSCW systems so the logs are confined to CSCW systems logs.

The CSCW systems that is used for this thesis is UniConnect, provided by the UCT. "UCT is a joint project of the University of Koblenz-Landau, IBM Deutschland GmbH and GIS AG. It was founded in 2010 as a university-industry cooperation project on the topic of Enterprise Collaboration Systems (ECS) providing expertise and services to academic institutions" [1]. The platform UniConnect is a highly collaborative platform with numerous

[1] https://uct.de/en/introduction/

integrated applications like wikis, blogs, forums, microblogs, chat, task management and libraries within communities. These applications provide plenty of user behaviour analysis and prediction possibilities, as user cooperation is involved in almost every one of them, which is one of the research goal for this work.

The UniConnect logs (Dataset) used in this research are from 2015. It is a Comma Separated value (CSV) file which is of 36 Mega Byte. The size of the data is not that big itself but, due to the high number of events (350,000), running complexity of used probabilistic model (RegPFA) becomes a constraint (section 4.1). The dataset contains twelve columns and only seven are used in this research as the meaning of some columns was not clear due to the privacy rules by data provider. Below is the brief description of columns used:

- **USER_ID:-** As clear from the name itself, it specifies the IDs of the users in the system such as user with ID 24576. However, user with this ID can be any user present in the system such as student, research associate or a professor.
- **ITEM_UUID:-** It describes the state where the users reside at a particular time such around the platform such as Homepage or Wiki, etc. For instance, user 17126 was on homepage at 1.33.43 o'clock and at 1.34.24 he was on communities page.
- **ITEM_TYPE_ID:-** It is the corresponding ID(Integer) for the Item name that is ITEM_UUID.
- **EVENT_NAME:-** EVENT_Name is the activity performed by the user on UniConnect platform at some specific time. Most of the values in this column are empty, as users are not performing activities all the time. For example, user with 27235 ID didn't perform any activity on 01-01-2015(Value=Empty) but on 05-01-2016 he uploaded the file on the platform(Value=File.uploaded).
- **EVENT_OP_ID:-** It is the corresponding ID(Integer) for the Item name that is EVENT_NAME.
- **COMMUNITY_ID:-** Every community across the platform have a Integer ID to which user belongs, but name of this communities are not provided in the dataset.

- **EVENT_TS:-** Date and time of every event recorded in the system.

Since, naming of the columns is always different for every dataset, calls for the need of a common naming convention that can be assigned to these columns, to use it with process mining algorithms.

2.2 Naming convention & tool used

Usually, all event logs are structured in a similar way and contain some standard elements for instance, events with case IDs. Hence, it can be asserted that CSCW logs also contain these elements as they are structurally similar to event logs (van der Aalst 2007). In literature there exist no consistent naming conventions for these elements but three most used conventios are:

- Process Mining Manifesto (Van Der Aalst & Adriansyah 2011)
- Naming in Fluxicon Disco (Günther & Rozinat 2012)
- XES 2.0 Standard (Günther & Verbeek 2009)

To enable productive collaboration with the tool used for this research, Disco naming convention is chosen as a viable option. Disco is a very powerful tool which take logs in CSV or extensible markup language (XMl) format and can convert them to eXtensible Event Stream (XES) or Multimedia eXtensible mark up language (MXML), which is the required input format for the most process mining algorithms including the RegPFA (i.e.,used for this work). Since, the data columns determine the possibilities of analysis so its important to have a naming convention for these columns. Disco requires minimum three attributes that is case_ id (Case identifier), activity, and timestamp. Hence, after uploading the raw log file (CSV or XML), one can simply configures which columns hold thecase_ id , timestamps, activity names and which other attributes should be included in the analysis as shown in figure 2.1. Meaning of all these attributes are as follows.

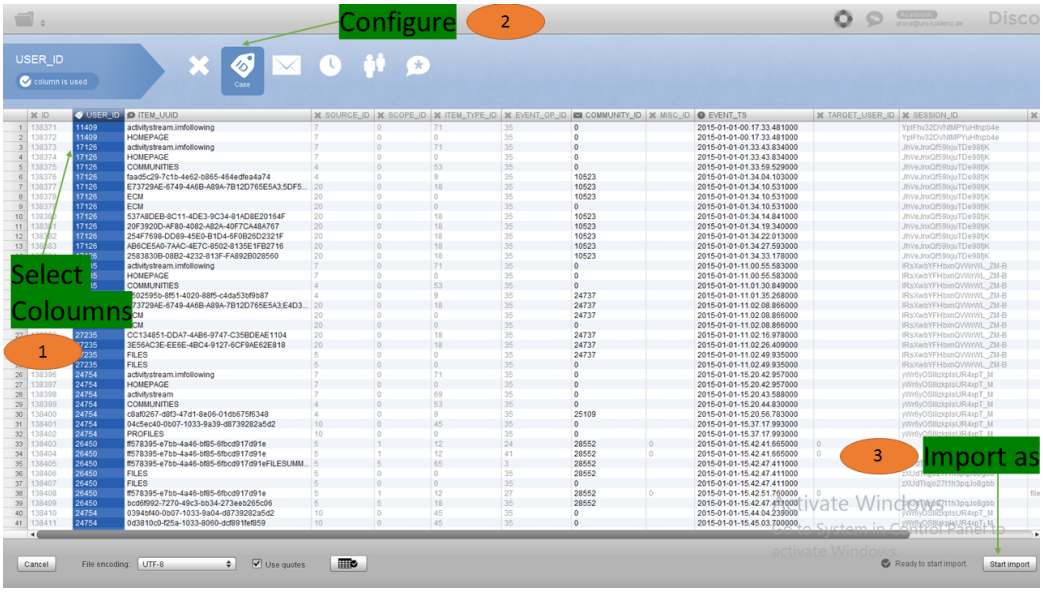


Figure 2.1: Step by step marking for conversion of CSV to XES in Disco (source:self-made, inspired by (Günther & Rozinat 2012)).

- **Case_ID:** Processes can have many instances and every instance is considered as one case with a certain ID called case_ id. Every step (event) in process must belong to case so that the scope of the process can be defined. Therefore, any column or more than one column can be assign as Case Id depends on the analysis goal or domain of the process. For example, in UniConnect logs Use_ID can be considered as case_ id

- **Activity:** Each step performed in every case is an activity. Each step can be represented by a name called activity. However, Activity should not be detailed as case level as every event in the process is not interesting for analysis. For example, a user uploaded the file can be considered as an activity but there will be some events where user will do nothing so at that time if the activity can be regarded as empty activity, which is not interesting for analysis can be filtered out beforehand.

- **Timestamps:** timestamp is when(Date&Time) a certain event is recorded. Timestamp is important to know when the activity and case was registered so that time related analysis can be performed for instance, which was the longest and shortest

case in the process history.

- **Other Columns:** Other columns are required to reveal the attributes of data in more detail and can be used according to the context of analysis. It is not important to use them but, if available its better to use to describe specific properties of process in more detail.

As soon as, the Case ID, Activity, Timestamp and Other Columns are applied to the holding columns, Disco provides various view tabs which can give a overview about the process. The major ones that are used for this research are as follows:

- **Process Map:** Disco create a process map by interpreting the sequences of activities where, path thickness and coloring of activities shows the main paths in the process flows as shown in figure 2.2.

Figure 2.2: Process map (flow) visualization In Disco (source:self-made).

- **Statistics:** This is one of the important aspect to analyze the process in detail such as total number of cases, activities and events, etc. Additionally, Disco also provides the individualistic statistics for instance, average time duration of cases, relative frequencies of individual cases and activities. Figure 2.3 shows a brief view of it.

- **Variants and Individual Cases:** This tab shows the cases in detailed information such as the events contained in a particular case. Inspecting individual cases is important to match the abnormal behaviour, if something similar appears in process analysis.

  Variant is the another striking feature of Disco. Variant can be seen as the common paths that some cases follows that is ”a specific sequence of activities” performed by different cases. They are really beneficial in some contexts, where main goal is to analyze the mainstream behaviour of process. For instance, to analyze the cases

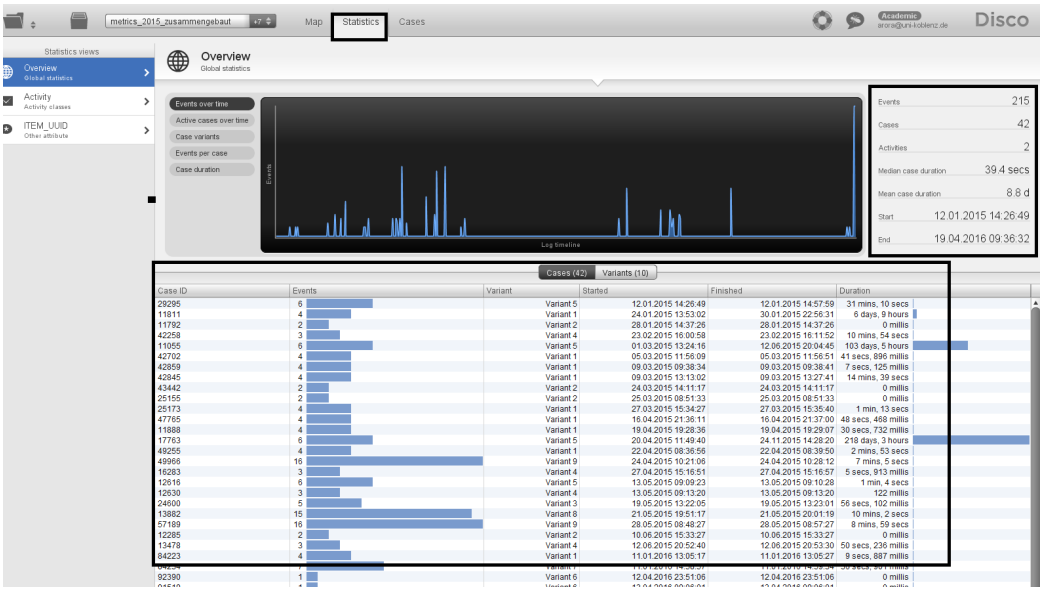


Figure 2.3: Statistics about process instance and individual cases (source:self-made).

whose sequence of activities is shared by at least two or three cases. Usually, there are only a few cases that follow the similar paths. Figure 2.4 shows both the variant and Individual Cases view.

- **Filtering:** Filtering is perhaps the most extraordinary feature provided by Disco, as many a times system logs contain lot of noise, namely incomplete cases, unusable cases,activities and events, etc, and for accurate process analysis it is important to filter them out. Figure 2.5 shows that Disco provides six different filter, which can be used in combinations and are as follows:

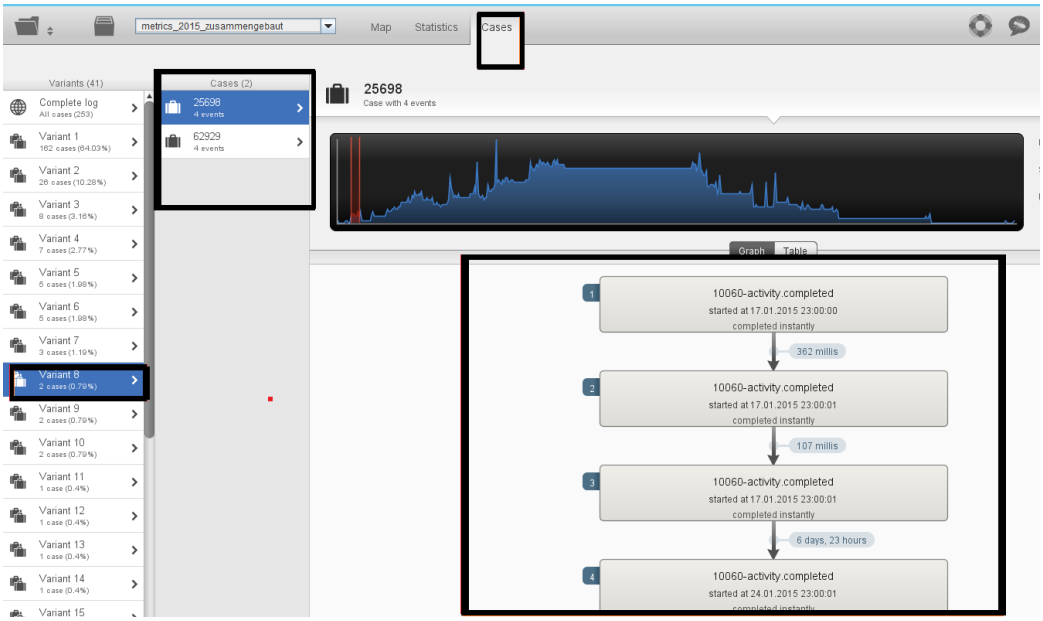


Figure 2.4: Variants containing different cases and events (source:self-made).

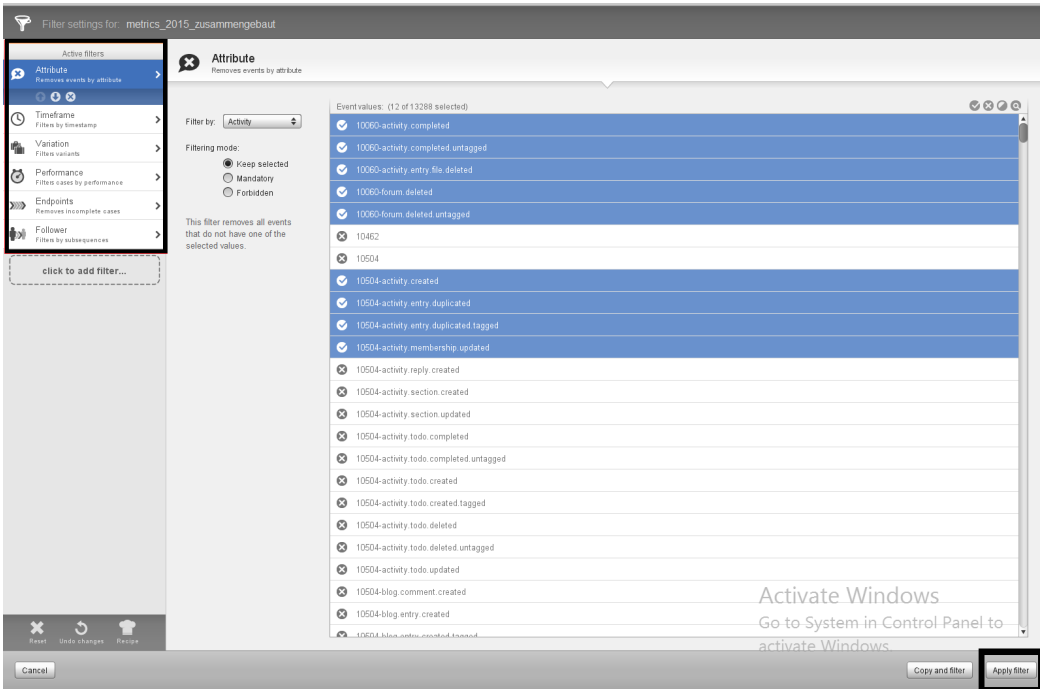


Figure 2.5: Filters available in Disco (source:self-made).

– *Attribute Filter:* Used to exclude the activities, cases or variants which are

not used for analysis.

- *Variation Filter:* Used to exclude the cases who do not share the common sequence of activities that is to find out the frequent path shared by cases
- *Performance Filter:* It allows to select the cases based on time duration of cases or the maximum and minimum number of events a case contains.
- *Timeframe Filter:* This filter helps to include or exclude the cases which are in or out of the certain timeframe as at times analysis is only confined for certain time period.
- *Endpoints Filter:* To select cases based on their start and end activities. For example,one can filter incomplete cases, or trim cases to cut out a part of the process.
- *Follower Filter:* Unlike the variation filter this filter provide the opportunity to filter the activity by follower sequence like direct follower or in some number of steps. However, this filter should only be used if the pattern in process are known. This work did not use this filter because there is no pattern information about the UniConnect logs is known beforehand.

2.3 Process mining

The log files discussed in section 2.1 and 2.2 are the input for the process mining step. VAN DER AALST ET AL. define process mining in their manifesto as ”techniques, tools and methods to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs commonly available in today’s (information) systems”(Van Der Aalst & Adriansyah 2011). Moreover, they define three types of mining in their process mining manifesto which are as follows:

- **Discovery:** The aim of process discovery is to synthesize a process model from an event log without having any a-priori information.

- **Conformance:** Conformance checking is the second process mining technique mentioned in the manifesto; it compares a given process model with an event log. This comparison enables to check if the real world execution of a process model (recorded in the log) conforms to the given model and vice versa.

- **Enhancement:** The aim of the third type of process mining is to enhance an existing process model by using information recorded in the log about the same process. In contrast to the conformance checking, which measures the alignment between model and reality, enhancement has the goal to extend or improve the given process model. The identification of bottlenecks by using the timestamps from the log is one example of enhancement.

The three aforementioned process mining types are explained in figure 2.6 in terms of input and output. The input for process discovery is an event log which contains sample executions of the process. Based on this event log, process mining synthesizes a process model which represents the process for example, in the form of a Petri net or a BPMN. The input for conformance checking techniques are an event log and the model representing the same process. By comparing the two inputs, conformance checking identifies differences and similarities between the given model and the event log. The third type of process mining, also needs an event log and a model describing the same process as input to deliver an improved or extended model as the output.
However, for the two step approach (probabilistic process mining and predicting user behaviour), that is used for this work, focus is on discovery aspect of process mining because to predict user behaviour in the second step of the approach, probabilistic process model is required. Many different algorithms solve the process discovery problem but only some advanced algorithms are able to provide real time information while, dealing simultaneously with other challenges that occur in real world logs such as noise, incompleteness, etc. Due to promising results on artificial logs and real world data this research used the the RegPFA algorithm developed by (Breuker et al. 2016). This probabilistic mining algorithm requires advanced parameter configurations to produce suitable results. Section

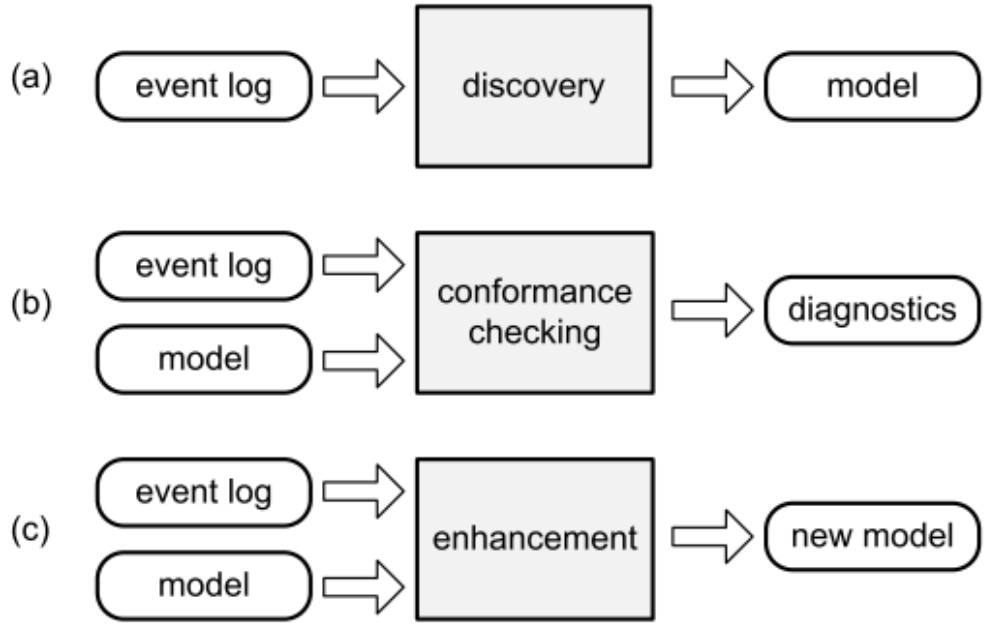


Figure 2.6: Types of process mining source: (Van 2011)

3.1 explains the basic idea of the RegPFA algorithm and exemplifies the setting of the parameters. The fundamental target conflict which arises in process discovery is to mine models which enable all variants in the log and at the same time do not allow new variants which are not contained in the log. To undertake this conflict Van (2011) four criteria, which are as follows:

- *Fitness:* the mined model should enable to replay all behavior observed in the event log.
- *Precision:* the mined model should not allow behavior completely unrelated to what can be observed in the event log.
- *Generalization:* the mined model should abstract/generalize from the given sample in the event log.
- *Simplicity:* the mined model should be as simple as possible.

Models with a good fitness are able to replay most of the traces from the log file. Secondly, model with poor precision is underfitted. For example, if a pattern is found in the model but does not exist in the log file. Whereas, generalization is related to the notion of overfitting. An overfitted model does not generalize enough because it is fitted only to a specific log file. The fourth quality criterion, simplicity means to search for the simplest process model which explains all behavior observed in the log file. This fourth quality criterion also increases the readability for humans. Section 3.1.3 discuss all these criteria in detail in terms of RegPFA.

Chapter 3

RegPFA and RapidProm

This chapter discusses an innovative process mining algorithm (RegPFA) that is used to achieve this thesis goals. The RegPFA stands for Regularized Probabilistic Finite Automata. RegPFA has been developed by Breuker et al. (2016) and its novelty in the process mining field resides in the underlying probabilistic approach. PFA is a well-known probabilistic model for problems similar to process mining that is why Breuker et al. (2016) applied this knowledge to process mining, specifically for process discovery problems.

3.1 RegPFA

This section is entirely based on the work of Breuker et al. (2016).

3.1.1 Probabilistic models

As this thesis aimed at discovering process models built from past data for the CSCW system in order to predict user behaviour, the probabilistic models also aimed in the same direction i.e., predicting how a specific running process instance may behave in the near future (Maggi et al. 2014). The main idea of a probabilistic mining algorithm is to have prediction capabilities by assigning certain probabilities to the process instances. The novel approaches of probabilistic mining alleviate large area of real time data analysis

subsumed under the term Big Data. For example, early warning system and anomaly detection (Chen et al. 2012). There are some predictive process mining approaches based on formal and logic models that describe a process as a set of all valid process instances to annotate the traditionally created process models with probabilities in order to have basic recommendations or predictions of likely future events (Van der Aalst et al. 2011). This type of predictions are quite similar to the field of grammatical inference as they build formal models that describe languages as sets of valid sentences (De La Higuera 2005). Nevertheless, these approaches often suffer from the log completeness i.e., if a variant is missing in input data will not be considered as valid process instance. On the other hand, some unwanted occurrences in the input data are considered as valid but in the form of noise. However, RegPFA is inspired from the approaches used in grammatical inference such as search engines, translation systems. Instead of defining the input data as a set of valid process instances, they use an underlying probabilistic distribution over all the process variants in order to manage the issue of *incompleteness* in input logs. Moreover, it also helps in addressing the problems from small learning data sets usually resulting in unreliable models, and from noisy data which can distort resulting models. Furthermore, another issue related to predictive modeling which RegPFA also covers is comprehensibility (Section 3.1.5). Predictive models often have bad visualization as they rely on several parameters which make the results elusive; hence, can be resisting when used for decision support (Provost & Fawcett 2013). Figure 3.1 shows an overview of RegPFA by showing the two components of it i.e., the RegPFA predictor that provides the predictive modelling functionality and the RegPFA analyzer that provides visualization and analysis. The next subsections of this chapter discusses the RegPFA components and its origins to better understand the advanced configuration necessary for the later usage in this thesis (Chapter 4).

3.1.2 Component first: RegPFA predictor

As discussed in previous subsection also that RegPFA uses the probabilistic logic models instead of adding each input variant as valid rule which is prevalent in formal models.

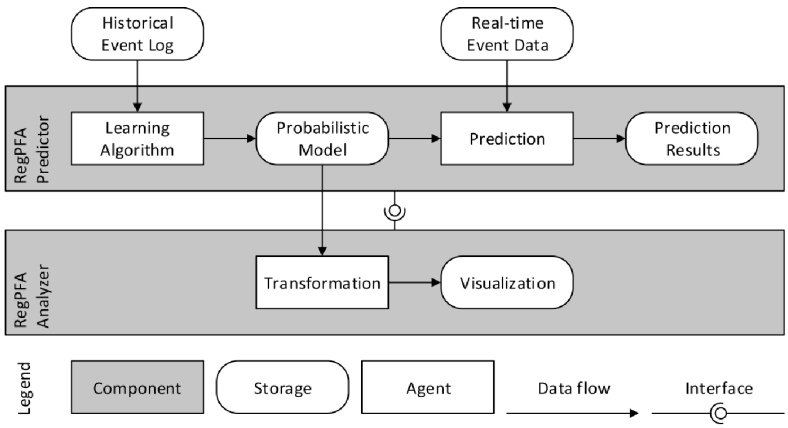


Figure 3.1: Overview of the RegPFA components (source:Breuker et al. (2016)).

The two most used probabilistic logic models are HMM and PFA and both are a kind of Bayesian network. In both the models next state depends on the previous state, that is, any state at time t depends on the previous state at time t-1. However, PFA not only consider the previous state into consideration but also the event that is taking place at time t-1 and this makes the PFA more similar to business process logic because they also assumes that a future state is dependent on previous state and the event or activity that is performed in between for example, in petri nets the next state is dependent not only on the previous state but also on the transition that is fired in between. On the contrary, there is one subtle difference also between PFA and business processes, that is, PFA can begin or end in any state as its values are chosen randomly by a probability distribution but, the business processes has designated start and end state; hence to fit the PFA to business processes researchers of RegPFA modified it by assigning a start and kill state in order to fix the start and kill event to only these states instead of a random state.

As PFA similarity to business processes makes it more suitable for the problem, its parameters of underlying distribution has to be estimated. The most common parameters estimation technique is Maximum-Likelihood-Estimate (ML) but ML can easily produced overfitted model as sometimes sample set is too small (Hastie et al. 2002). For example, if a dice is thrown one time and hundred time where in both cases six appears all the

time, then the maximum likelihood delivers a probability of one in both the cases. Hence, when maximum likelihood is applied to the case where dice was thrown only one time it will produce a model which is highly overfitted. To overcome this problem Breuker et al. (2016) used the Bayesian regularization as business processes often have incomplete data which helped in overcoming the problem of overfitting by adding the expected value to the probability; hence, extreme values will always be suppressed. This bayesian regularization in RegPFA named as *pseudo-obervations*. As the Bayesian regularization was applied to PFA, an underlying probability distributions for the parameters of distribution have to be chosen and for that Dirichlet distribution was chosen by Breuker et al. (2016). However, to do a maximum likelihood for the parameter estimation, Expectation maximization (EM) algorithm was a preferred choice as it was the state of the art to estimate the parameters of probabilistic distribution with unobserved variables(Dempster et al. 1977). EM algorithm starts with a randomly defined initial parameter configuration and iteratively changes the parameters to improve quality criteria (subsection 3.1.3). The EM algorithm stops when the improvement between two iterations falls below a certain threshold or in the RegPFA implementation when a maximum number of iterations is reached. However, EM requires a certain number of inputs to estimate the parameters for the probability distribution and these inputs are the data i.e., used to train the model, number of events, regularization strength and model state space. The EM algorithm can only estimate the parameters for the probability distribution on input data but it does not know beforehand how many states best represents the input data or whether the regularization improves the quality of the mined model and how strong the regularization should be. Hence, for every combination of regularization strength and number of states also called as grid search EM will discover a process model. As EM converge towards a local optimum therefore, in search of best fitted process model EM has to run in various iterations with different initial values of the number of states and regularization strength combination. Next subsection discusses the criteria to find the best fitted process model from the all the process model generated by RegPFA.

3.1.3 Model scorers (KPIS)

The RegPFA calculates probabilistic models with different variations of grid search i.e., varying the number of states and the regularization strength. However, the EM algorithm also produces many models with the aim of iteratively improving the quality of explaining the input data. Hence, to calculate the best fitted model three quality criteria or model scorer was used by the researchers of RegPFa which are as follows:

- **Akaike Information Criterion (AIC):** The AIC relates the likelihood achieved in EM step while estimating the parameters to the amount of parameters needed and if there are additional parameters need for the explanation of data it penalizes them.
- **Heuristic Information Criterion (HIC):** This parameter was developed by the researchers of the RegPFA as an advancement of AIC. HIC punish the parameters after a certain number because in business processes most states have only few types of events; hence when the process runs the states which can process that events will also be less, making many model parameters to zero. That is why they modified the AIC by punishing the parameters above a certain threshold named as HIC. Moreover, for this thesis HIC was also chosen as the quality criteria because CSCW system have less structured processes leads them to record all the various types of events possible (van der Aalst 2007). Furthermore, in UniConnect logs it was seen the varying range of events was quite high.
- **Cross-Entropy (CE):** Another quality measure that was provided by the researchers of the RegPFA is CE, which tests the probabilistic model against a test data set that was not used for learning.

Lastly, in all the model scorers criteria of evaluation was *lower the model score is, better the model is.*

As discussed in section 2.3 Van (2011) also provide four quality criteria (i.e., fitness, precision, generalization and simplicity) in order to avoid overfitting and underfitting in the process model discovery. However, for this thesis it was important to configure the

RegPFA algorithm towards fitness and simplicity because a high fitness means that the probability for false negatives during the user behaviour prediction will be low. In this context a false negative means that a user behaviour can be found in the UniConnect but not in the mined process model. On the other hand, user behaviour has to be predict manually from the mined process model so it has to be simple and that is why different iteration of every model (section 4.2-4.8) was performed to achieve a process model that is simple and best fitted. On the other hand, it was also important to avoid underfitting i.e., with low precision because they allow pattern that can be found in the model even though in the logs no such violation exist. Lastly, the probabilistic mining approach ensures generalization to some degree inherently.

3.1.4 RegPFA parameters

As discussed in subsection 3.1.2 there were several parameters that has to be adjusted before starting the RegPFA, this subsection discusses all those parameters. In previous subsections RegPFA parameters meant the parameters of the distribution i.e, the input parameters but from now on RegPFA parameters will mean the RegPFA configuration (such as states in grid search, EM iterations) not the parameters that has to be estimated. *RegPFA Configuration* was replaced by the *RegPFA parameters* because configuration word was already used for Disco Configuration hence to avoid confusion in the next chapters between Disco configuration and RegPFA configuration, RegPFA parameters will be used instead. The parameters that has to be initiated before starting the RegPFa are as follows:

- **Prior strength (Regularization strength):** It determines how many pseudo-observations are added based on the number of events in the log (0.1 means 10% of the number of events in the log). Since, it was not known a priori which regularization strength explains the input data best, the grid search had to try a few to find the best one according to the scenario.

- **Minimum and Maximum States:** They set the start and end point of the grid search as these values are unknown and can not be optimize; hence, their values

has to be decided by using the simple heuristics that is depending on the number of cases, activities and events.

- **Maximum EM iterations:** IT is one of the stopping criterion for the EM algorithm. Usually, the EM algorithm should converge or fall below a certain threshold of improvement (EM threshold). Hence, stopping the algorithm after an arbitrary number of iterations was avoided.

- **EM threshold:** IT is the second stopping criterion and the effective one. The algorithm compares the log-likelihood of a new probabilistic model to the previous one and determines whether the improvement fell below the threshold.

- **Tries per run:** IT relates to the repetitions of the exactly same parameter setting. This was recommended because the EM algorithm can run into local maxima depending on the randomly initialized starting position.

- **Model selection criterion:** IT can be one of the AIC, HIC or CE. But for this thesis HIC was chosen (3.1.3).

- **Pruning ratio:** IT determines the threshold which minimum probability a transition must have to be included in the transition system or later on in the Petri net. The pruning ratio is an attempt to generalize this configuration value as it asks how many times more unlikely the probability of a transition is allowed to be compared to a base probability which depends on the specific log.

3.1.5 Component second: RegPFA analyzer

As discussed in subsection 3.1.2 that RegPFA analyzer handles the issue of comprehensibility and human readability in probabilistic models, this subsection explain how RegPFA manage these potential downsides in probabilistic models. Process mining outputs are usually in the form of BPMN or Petri net but probabilistic models can not be represented like that. However, this weakness was overcame by RegPFA in the form of RegPFA analyzer which transforms the probabilistic model into a transition system, which again can

be transformed into a Petri net. Additionally, this visualization part of the RegPFA is not directly related to the predictive process mining approach. Rather its a functionality to visually and manually evaluate the model quality or to perform analyses on the mined model for example, in case of this thesis it is used to predict user behaviour in UniConnect platform. One disadvantage of the RegPFA analyzer was that it could not include the probability on the transition systems. However, how transition system will look like depends on these probabilities because a threshold (inverse of pruning ratio) (subsection 3.1.4) determines which minimum probability a transitions must have to be included in the visual representation. This threshold was important since a probabilistic model connects every state with every other state via each possible event and associates probabilities to these transitions including too many transitions with low probabilities would overload the visual model if it is used for manual analysis such as in case of this thesis. This threshold needs some experimentation because infrequent variants are usually interesting. But, since for the predictive applications RegPFa used the Bayesian regularization, it assigns probabilities greater than zero to all transitions even though this variant was not included in the learning data. Therefore, lowering the threshold leads to the inclusion of pseudo-variants which could distort analyses and produce underfitted models. Hence, selection of threshold value was quite important for the correct predictions of user behaviour.

3.2 Rapid Miner and RapidProm

The RegPFA is implemented as a plugin in ProM. Where ProM is a process mining tool developed by the (Van Dongen et al. 2005). However, running the RegPFA in ProM was very time consuming and not efficient. The manual steps required in ProM make it a cumbersome. Moreover, ProM only delivered the results with best value of the model scorer which can be pruned by pruning ratio set beforehand. If another pruning ratio has to be applied the whole results has to be run again which is very time consuming as the runtime of some models were even upto two weeks (section 4.3-4.8). Moreover, if some other parameters has to be looped such as EM threshold then the whole process has to be

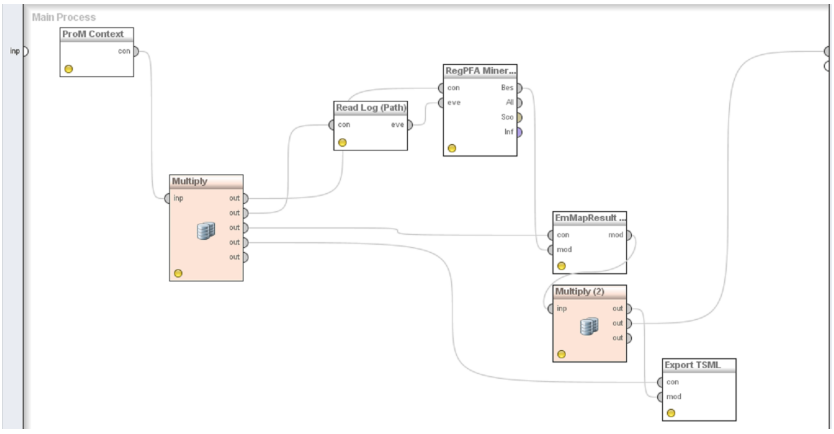


Figure 3.2: Workflow in RapidMiner (source:self-made).

start again.

But ProM framework has a a plugin for the ETL tool RapidMiner with the name RapidProM. RapidMiner is an environment to execute multiple data manipulation steps and workflows within the various domains such as machine learning and data mining (Land & Fischer 2012). RapidMiner is based on client and server model where the server part is delivered as a software as a service (SaaS). RapidMiner works on the basis of workflows and provide visualizations as well and RapidProM uses that efficiently (Mans et al. 2014) and allows to call the functions of ProM plugins by workflows built and executed in RapidMiner. RapidProM has the capability to build and execute workflows within the process mining domain. A typical workflow is shown in Figure 3.2. First a log is read, afterwards a process mining algorithm (e.g. RegPFA) is executed and provides a transition system as the result. Workflow has several building blocks called as operators. Each operator perform a specific fucntion for example, the RegPFA miner it can mine like normal RegPFA miner in ProM but also has the capability to loop the several parameters or execute it with different parameter settings on different log files. An operator in Rapid miner have input and output ports where output of one operator can be used as input for another operator if required as shown in figure 3.2

3.2.1 RegPFA operator

There were two types of RegPFA operator availaibe in RapidProm plugin of Rapid miner. First one was the *RegPFA miner* which provide transition system as an output while the another one called *EM Map Result* which does not provides a transition system but data structure that we call *EM Map Result*. An EM Map Result represents a not yet pruned transition system, providing the user an option to prune this result with as many different ratios to find a suitable pruning ratio and does not need to re-initiate the whole mining algorithm again. To prune with many different ratios a slider was provided called as *live pruning*. Using this slider the pruning ratio can be changed on the fly and the pruning of the transition system happens immediately. This option is also useful to find a suitable value for the ratio without the need to loop several values in the workflow process. Moreover, this operator does not provide only best model but also has an output port called *all* which delivers all the models with all the model scorers which can be sorted as well to find the best model. This best model can be run again with different parameters with the loop. Hence, RapidProM was really useful and effecient for this work. It was possible to build workflows that try several mining parameters and mine a batch of log files. This makes the execution of experiments less time consuming. The workflow that was used for this thesis is discussed in detail in section 4.2.

Chapter 4

Outcomes and evaluation of created models

This chapter illustrates seven different models. Three of them have different dataset configurations (such as case_id, activity, etc.) and variable parameters (such as minimum states, EM iterations, etc.) of RegPFA algorithm. Another three models have the respective configurations but different parameters, as they run the algorithm only for the most frequent paths. The model fifth runs the algorithm for the multiple activities performed by the user (i.e., combining two columns as activity ID). However, the model zero (Naive model) runs the algorithm without any filtering, that is, using all the events present in the dataset. The rationale behind multiple models is to discover the suitable configurations and parameters of the UniConnect (CSCWC system) logs and RegPFA algorithm respectively in order to explore the possibilities of predictive process modelling for CSCW system (more specifically, UniConnect) and to predict the user behaviour from the corresponding mined probabilistic model.

4.1 Model 0: naive model

This model was used to have a naive idea about the complexity and range of dataset in terms of runtime and primary configuration (no filtering of cases, activities and events). Primary configuration of all the other six models have around 1000 to 1200 cases, 300000 to 350000 events and 200 to 400 activities. These huge numbers in configuration leads to an inadmissible run time of RegPFA. For example, table 4.1 shows the parameter used for a naive run that had 348513 events classified in 1100 cases and 241 activities.

RegPFA parameters	Statistics
Minimum states	20
Maximum states	60
Regularization strength	0.3
EM iterations	100
EM thrushold	0.001
Number of tries	1

Table 4.1: Parameters for naive run.

As it can be seen from the table 4.1 there were only minimal parameters that were used for the naive run i.e., a single try for EM convergence in 100 iterations and a only one regularization strength of 0.3 for the grid search of all the process models between 20 and 60 states and yet, the runtime of RegPFA was one month and algorithm was still running. This untenable runtime of RegPFA and huge numbers in configurations calls for the need of deeper inspection and filtering of dataset according to the scenario. Next subsection discusses the methodology that is used for all the six models to overcome the hindrances in the naive run

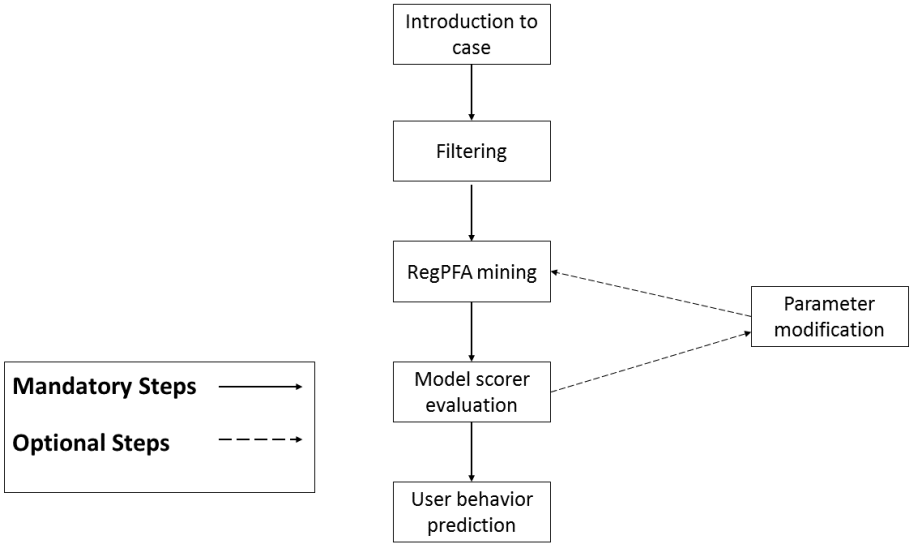


Figure 4.1: Methodology used for models (source:self-made)

4.2 Models methodology

Figure 4.1 shows the methodology used for all of the ten models where dotted arrows indicate that the corresponding steps are optional. The following subsection depicts the different sub parts of methodology.

Introduction to case: It describes the scenario for the model by explaining the Disco configuration (such as case_id, activity, etc.) for the corresponding UniConnect dataset columns. Additionally, insights into the statistics of the model are provided, for instance, number of cases, events and median case duration. Furthermore, it also checks if there are some extreme or exceptional behaviour in the dataset for the particular configuration selected.

Log filtering: Log filtering was important to have a summary about the events, cases and activities as well as putting vital data in analysis; for instance, it is irrational to include the cases in analysis which are still running (Van der Aalst et al. 2009). Depending upon the scenario, filtering is an optional step for the logs with structured processes, such as event logs. In contrast, the less structured processes of CSCW system record almost every

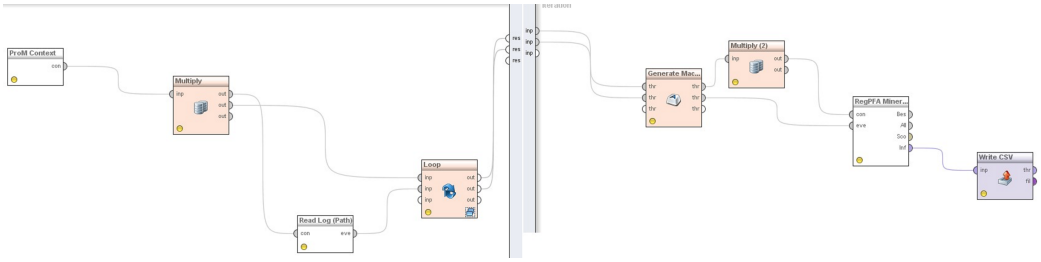


Figure 4.2: Rapid miner process workflow used for all the models (source:self-made).

event, making the number of events quite huge. Hence, log filtering becomes an essential step for CSCW systems logs as many events are not even relevant for the analysis, for instance, empty activities (i.e., user is ideal). Consequently, it conceptualizes the fact that UniConnect logs also require filtering as they belong to the category of CSCW systems. The filtering of every model is explained in its description as it depends on the scenario and corresponding configuration (such as case_id, activity, etc.) selected.

RegPFA mining & parameter modification: To mine the probabilistic model Rapid Miner provides two operators namely, emMap miner and RegPFA miner but for all the seven models emMap miner was preferred as a better choice due to its advantages over the RegPFA miner discussed in section 3.2.1. The Rapid Miner workflow that is adopted for all the models is shown in figure 4.2, and below is the brief description of all the operators used in workflow.

- **Prom Context:** This operator provides the prom object that is needed for every prom operator, such as read log, RegPFA miner, etc.
- **Multiply:** This operator only transfers the output of one operator to another as-is.
- **loop:** This operator loops the parameters of another operator when used with macro and in this case it was used to continue the grid search because in some process models grid search was finished own its own.
- **Generate macro:** This operator can be used to calculate new macros from existing macros. A macro can be used by writing %{macro_name} in parameter values of

succeeding operators of the current process. In this workflow it was used as macro for maximum and minimum states in RegPFA Miner (Em Map Result) miner and the value of the this macro was coming from the count of the loop operator behind it.

- **RegPFA Miner (Em Map Result):** As already discussed in section 3.2.1, this operator provides the not yet pruned transition system (probabilistic process model).
- **Pruning Ratio slider:** It provides the user an option, to prune the transition system produced by the Em Map Result operator with the help of a scaled slider from zero to one. Optimal transition system can be found out by selecting various pruning ratios along the slider.

Additionally, every model consist of various iterations of RegPFA parameter (such as, minimum state, EM threshold, etc.) modifications in order to discover a optimal process model. The iterations were continued until the optimal process model was not achieved.
Model Scorer Evaluation: AIC, HIC and CE Test are three main knowledge performance indicators (KPIs) available in RegPFA but for this thesis HIC was selected as the model scorer (section 3.1.3) for the best fitted process model with corresponding parameters (such as number of states, EM threshold, etc.). The best fitted model should have the lowest HIC score. Every model in the following sections implements the plot of HIC with the number of states that a process model contain. However, model first implement the plot of all the three KPIs to justify why HIC was selected over the AIC and CE test (section 3.1.3 and 4.3.3).
User behaviour prediction: This section predicts the user behaviour from the best fitted probabilistic process model (transition system) produced by the RegPFA. For instance, the activities that will be performed by user in near future or the communities that user will join. However, user behaviour prediction was a manual process so the probabilistic transition state diagram of best fitted process model was trimmed to different pruning ratios until it was possible to analyze the transition diagram manually. Pruning ratio was selected from slider in such a way that there is minimum loss of information and manual

analysis of transition state diagram can be done simultaneously.

4.3 Model one: predicting users activity

4.3.1 Introduction to the case

This model is focused on predicting the very basic behaviour of users that is the series of activities performed by users on the UniConnect platform. Correct prediction of activity sequence requires the selection of appropriate columns from the dataset. Thus, User_ID was taken as case_id, Event_name as activity_ID and Event_TS as the time stamp. Event log for this model have 348,513 events which were classified in 241 activities. The 1044 variants were executed in 1139 cases. This illustrates the two very exceptional characteristics of the log file:

- *The relative frequency of empty activities was very high, 77.9 percent to be precise as shown in figure 4.3*

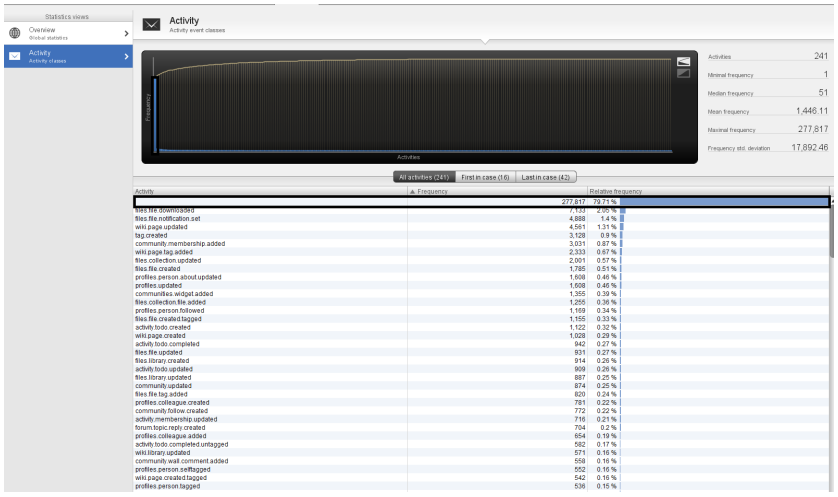


Figure 4.3: Empty activities indicated by a black rectangle (source:self-made).

- *There were huge number of cases with very few events and vice-versa as shown in figure 4.4*

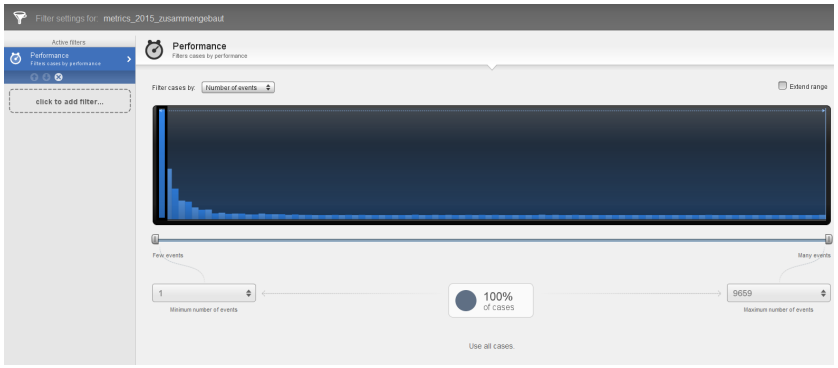


Figure 4.4: Cases with few events indicated by a black rectangle (source:self-made).

However, in terms of UniConnect the black rectangles in figure 4.3 indicates that most of the time users were ideal; in other not performing any activity. On the other hand, the black rectangles in figure 4.4 points out that there were many users who performed fewer events and the other way round. This brief view on the log file of UniConnect shows the complexity and the range of the dataset while illustrating, how necessary it was to filter (section 4.3.2) the data to achieve a sustainable quality of information.

4.3.2 Log filtering

To filter the extreme behaviour mentioned above, Disco performance and attribute filter was used, where the former has the capability to filter the cases on the basis of case duration and number of events, the latter filters out on the basis of particular cases or activities. Nevertheless, the interesting point was the order in which both the filters were applied as shown in tables 4.2 and 4.3

Statistics after	**Events filtered first**	**Activities filtered second**
Number of cases	613	405
Number of events	326402	68194
Number of activities	240	240
Mean case duration	21.5 weeks	47.2 days

Median case duration	25.5 weeks	98.3 days

Table 4.2: Performance filter applied first that is filtering events in the first place.

Statistics After	Activities filtered first	Events filtered second
Number of cases	987	769
Number of events	70696	54906
Number of activities	240	233
Mean case duration	69.4 days	16.8 weeks
Median case duration	20.8 days	22.7 weeks

Table 4.3: Attribute filter applied first that is filtering activities in the first place.

As it can be seen from the table 4.2, where the event filter was applied first, there was a drastic drop in number of cases (i.e., losing information) rather than events but as soon as the activity filter was applied, the number of cases were the same but there was a huge drop in the number of events. Whereas, in table 4.3 where the activity filter was applied first, behaviour was completely the other way round as compared to table. Two important deductive points from filtering were:

- *How the order of the filters can prevent the amount of information loss as when event filter was applied first it filters most of the cases or in other words it deletes most of the users from the analysis. Besides, the main goal is to predict user behaviour.*
- *Some prior knowledge about the logs is also important as in this case, it was known that empty activities did not contribute to analysis. Hence, there events can be deleted as higher number of events lead to longer runtime of RegPFA (section 4.3.3).*

Therefore, in order to have minimum information loss activity filter was applied first followed by event filter, remaining with 73.5% of cases, 97.08% of activities and 15% of

events. Still, the runtime of RegPFa mining was two weeks and three hours (section 4.3.3). To produce a single digit runtime, further filtering of logs were required; certainly, this results in some more information loss but the goal is to search for such an exceptional behaviour where the information loss is minimal and a good balance can be achieved between runtime and information loss. Thus, finding out whether or not another exceptional behaviour was important. Inspecting the statistics from table 4.3, difference between the mean and median case duration can be clearly seen, indicating the presence of outliers. More specifically, there are cases with very low duration and vice versa. According to Conforti et al. (2015) the filtering of outliers always improves the quality of process model discovery along the quality criteria suggested by Van (2011) in the process mining manifesto and these criteria are fitness, appropriateness, simplicity and generalization. However, it depends upon the scenario which criteria should be given importance but, as already discussed in section "still to look" RegPFA was configured more towards fitness to avoid false negatives. Thus, to have a appropriate sample for the analysis this behaviour was trimmed using the performance filter of Disco. Statistics after removing the outliers are given in table 4.4.

Statistics after performance filter based on case duration	
Number of cases	673
Number of events	33405
Number of activities	228
Mean case duration	19.1 weeks
Median case duration	16.8 weeks

Table 4.4: Statistics after the performance filter i.e., to normalize case duration.

It is clearly visible from the table 4.4 that the difference between mean and median case duration was reduced to three weeks in comparison to six weeks when this filter was not applied. Whereas, there was some loss of information in terms of number of cases but this loss is acceptable when compared to the reduction in number of events which is directly

proportional to the runtime of RegPFA algorithm. Next subsection explains, in detail, how filtering at different levels affects the run time of RegPFA algorithm.

4.3.3 RegPFA mining & parameter modification

As discussed in the previous subsection how filtering techniques at different levels play a vital role in log quality of this particular model, this subsection explains how various iterations of RegPFA parameter modification (such as Maximum and Minimum states, EM iterations, etc.) together with filtering not only leads to achieve a better runtime of algorithm but also a better fitted process model.

First iteration of parameters that were used with the filtered logs (i.e., filtering the empty activities and cases with very few events (table 4.3)) is shown in table 4.5. The filtered logs contain 73.5% of cases, 97.08% of activities and 15% of events in comparison to the 100% of cases, activities and events used in the naive run of RegPFA (section 4.1) where it was running forever.

RegPFA parameters	**statistics**
Minimum States	50
Maximum States	100
Regularization strength	02,0.3,0.5
Number of tries	3
Model Scorer	HIC
EM iterations	100
EM thrushold	0.001

Table 4.5: Parameter details for the first iteration of model one.

However, the runtime with filtered logs and the parameters (table 4.5) was around twelve days and according to the HEC model scorer, model with 73 states was the best model. Justifiably, runtime was improving in comparison to the naive run where it was running for a month and still not ending. But in this run, EM was never converging probably

because of the too less EM iterations

To overcome that in the second iteration, EM iterations was increased to 500 and the number of tries was set to one as the number of tries was not improving the model scorer rather increased the runtime. Even after the single try runtime was 13 days, that is, approximately only one day more than the previous run. The higher number of EM iterations leads to longer runtime because all the consecutive states between minimum and maximum states tries to achieve a local optima in 500 iterations as compared to 100 of the previous model. Moreover, the EM was converged in two process models, at first, in process model with 68 states and after 468 EM iterations and secondly, after 457 EM iterations in process model having 74 states in it. Whereas, the best process model according to the HIC model scorer had 66 states in it without having EM convergence. Table 4.6 shows the comparison between the two models i.e., model with 100 and 500 EM iterations respectively.

RegPFA parameters	**EM iterations 100**	**EM iterations 500**
Minimum states	50	50
Maximum states	100	100
Regularization strength	02,0.3	02,0.3
Number of tries	3	1
Model scorer	HIC	HIC
EM thrushold	0.001	0.001
Runtime	12 days	13 days
Best Model with states	73	66

Table 4.6: Comparison of the two best models with EM iterations 100 and 500.

Hence, it can be argued that 500 EM iterations and 0.001 EM threshold was optimal for a good fitted model. However, to find out the best fitted model in terms of number of states space still existed. Thus, in the third iteration a new parameter for minimum and maximum states selected that is 100 and 150 respectively by keeping the other parameters

similar to the one given in table 4.6. Table 4.7 shows this parameters in detail.

RegPFA parameters	statistics
Minimum states	100
Maximum states	150
Regularization strength	02,0.3
Number of tries	1
Model scorer	HIC
EM thrushold	0.001
Runtime	Never finished

Table 4.7: Parameter details for the third iteration of model one.

This increase in maximum and minimum states lead the algorithm to run forever, after running for two weeks there were only four process models achievable i.e., models with 100, 101, 102, 103 and 104 states in them.

Hence, it can be contended that the best fitted process model for this configuration is model with 66 states in it with a regularization strength of 0.2 while having a EM threshold and iterations of 0.001 and 500 respectively. Table 4.8 showed the detailed view of all the other parameters such as number of tries and pseudo observation.

RegPFA parameters	statistics
Minimum states	50
Maximum states	100
Regularization strength	0.2
Number of tries	1
Model Scorer	HIC
EM thrushold	0.001
Runtime	13 days

Table 4.8: Parameter details for the best fitted process model.

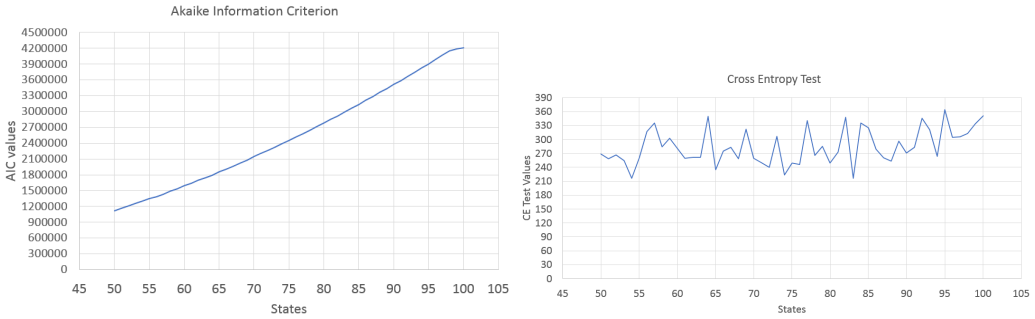
It can be clearly seen from table 4.8 runtime of RegPFA was around 13 days and to reduce the runtime to single digit further filtering of logs was done as mentioned in section 4.3.2 and the statistics related to it can be seen from and serve as the input logs for the fourth iteration.

Since, in fourth iteration the parameters of the best fitted model (table 4.8) before the case duration filtering was known, these parameters were used as the basis for the next run because the difference in number of events and cases, before and after the case duration filtering was not that huge, when relatively compared to the original number of event (348513) and cases (1044). After running the algorithm with the parameter mentioned in table 4.8 runtime was reduced to seven days and process model that have 95 states in it have the lowest HIC score of 38601. However, similar to the second iteration in this situation as well, the EM was not converging for the process model with lowest HIC score but for the other process models containing 64 and 84 states whereas, EM was converging after 474 and 486 iterations.

Hence, for the model first configuration, the best fitted process model that justified both the runtime and RegPFA parameters contains 95 states in it while having a runtime of seven days. Table 4.9 shows the RegPFA parameters for the best fitted process model for the first model, whose configuration is presented in table 4.4

RegPFA parameters	**statistics**
Minimum states	50
Maximum states	100
Regularization strength	0.2
Number of tries	1
Model Scorer (HIC)	38601
EM thrushold	0.001
Runtime	11 days

Table 4.9: Parameter details for the best fitted process model of model one.



(a) AIC for the best fitted process model (b) CE for the best fitted process model .

Figure 4.5: KPIs for the best fitted process model (source:self-made).

As already discussed in model methodology and section 3.1.3 that why HEC fits better and chosen as the model scorer for this work therefore, only HEC graphs were discussed in detail. However, for this model (i.e., first model) all the three KPIs (HIC, AIC, CE) were discussed, in order to have an example for the same.

The plot in figure 4.5a shows shows the AIC value on the y-axis compared to the number of states of a mined model on the x-axis. To recapitulate, a lower KPI value expresses a higher quality of the model (section 3.1.3). However, in case of AIC the value was just increasing for all the process models from 50 states to 100 states. Hence, it was inconclusive to say which process model is optimal. On the other hand, plot in figure 4.5b shows shows the CE test value on the y-axis compared to the number of states of a mined model on the x-axis. It can be clearly seen from the model that CE perform better as the increase was not exponential but still the trend was quite fluctuating.

Lastly, the figure 4.6 shows the plot for HIC. It can be seen from the figure the HIC score was shooting in the beginning but as the number of states was increasing HIC had some spikes with a decreasing tendency. There was a sudden downfall before the the lowest HIC score of 38601 for the process model that had 95 states in it. However, before achieving the lowest score there was a sharp spike in HEC score hence it can be argued that if the states will increase its highly likely that HIC will increase with some small spikes as the gap between the downfall and rise of HIC value was the biggest at state 95. Next subsection tries to predict the user behaviour from the discovered best fitted process

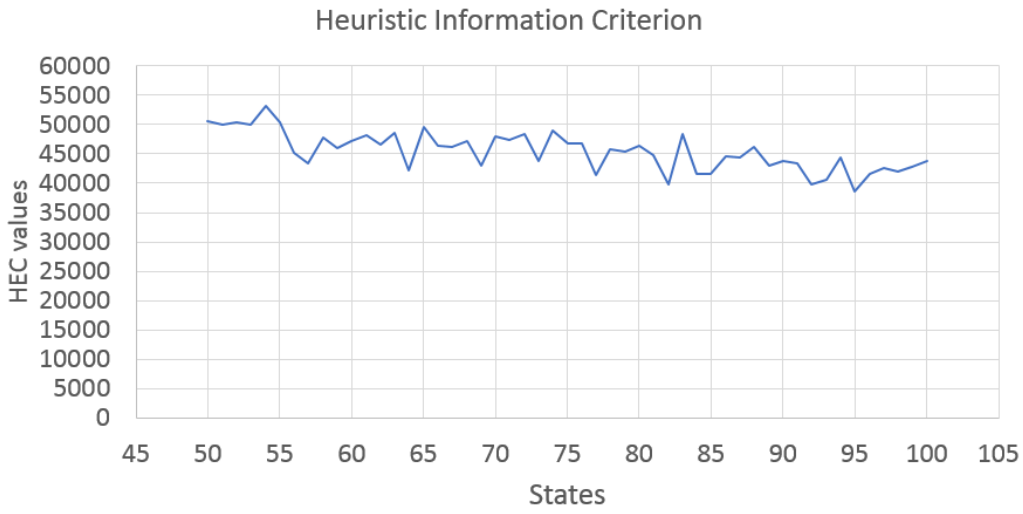
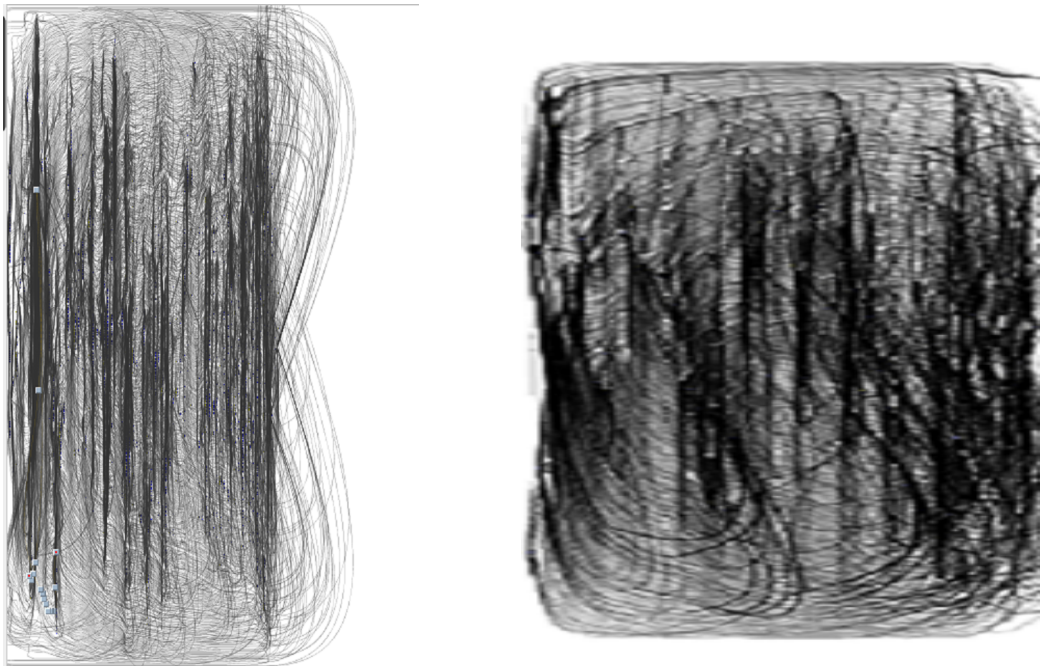


Figure 4.6: HIC for the best fitted process model (source:self-made).

model (transition system) by having multiple pruning ratios.

4.3.4 User behaviour prediction

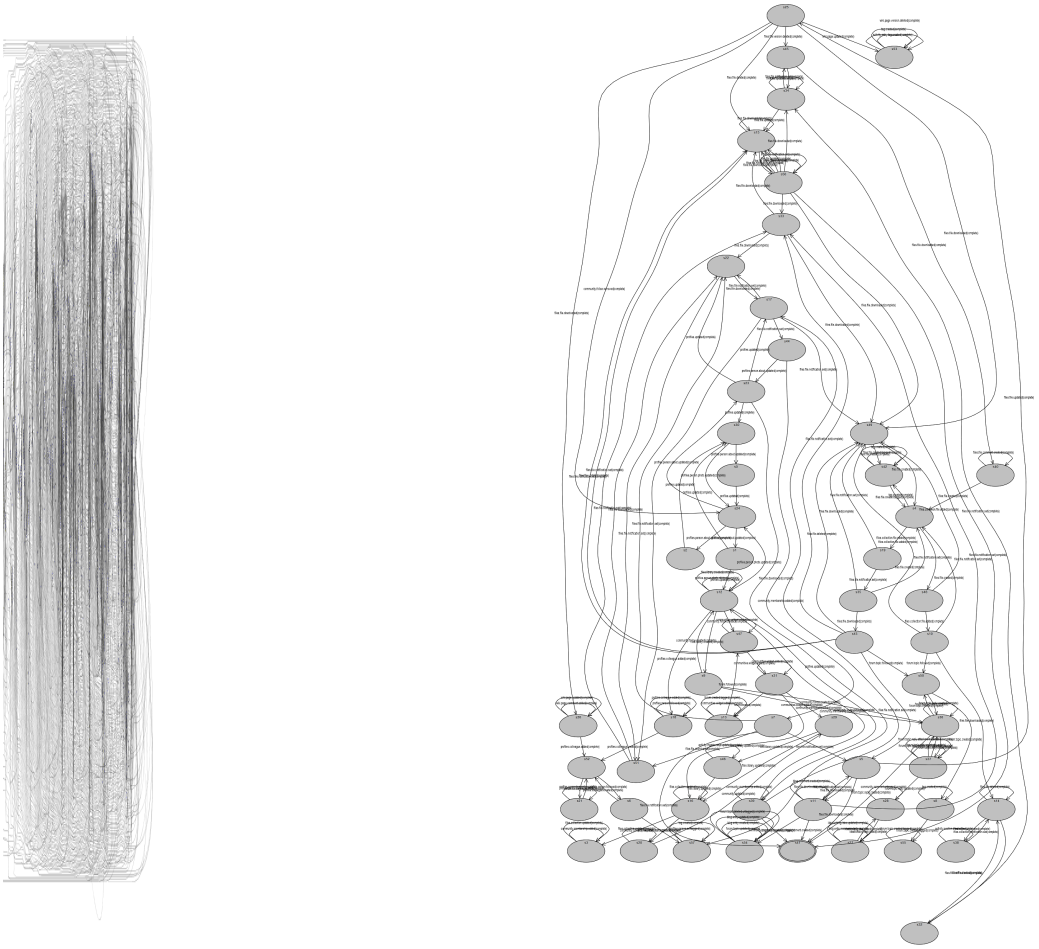
To predict the activities that will be performed by the user in near future, analysis of probabilistic transition system produced was critical. Evidently the transition system (process models) even with very low pruning ratio was tangled like a yarn and to predict user behaviour manually from the transition systems produced also proved insurmountable even after very low pruning ratios as shown in figure 4.7a and 4.7b with 0.35 and 0.1 pruning ratios. Hence, lowering of pruning ratios were required in order to predict user behaviour from the probabilistic process model produced. However, as discussed in models methodology (section 4.2.5) that lowering of pruning ratio should be done by maintaining a balance between information loss and visualization in order to find the optimal pruning ratio. Thus, with the help of slider pruning ratio was further reduced to 0.009 and 0.0009 as shown in figure 4.8a and 4.8b. The pruning was directly reduced from 0.009 to 0.0009 because in between that pruning ratios it was still not possible to visualize anything. However, at 0.0009 the process model was moderately visible but, still can not be analyzed manually. Finally, after lowering the pruning ratio to 0.0004 as shown in figure



(a) Insignificant process model with 0.35 pruning ratio.

(b) Latent process model with 0.1 pruning ratio.

Figure 4.7: Camouflaged visualization for model one (source:self-made).



(a) Spaghetti process model with 0.009 pruning ratio

(b) Moderately visible process model with 0.0009 pruning ratio.

Figure 4.8: Unobservable process for model one (source:self-made).

4.9 it was possible to analyze the process model in order to predict the activities that will be performed by the user. At this stage due to very low pruning ratio only four activities were left out of 228 which also increases the chances of an underfitted model. The three activities are as follows:

- *Updation of profile*
- *Updation of persons profile*
- *Updation of profile picture*
- *files library creation*

Starting from state S0 and follow the outlinks from that state to the state where there are more outlinks there were two highly likely predictions possible about the activities that user would perform:

- *Its highly likely that users will update their profile followed by updating the profile picture and finished by updating their information about them*
- *Its also highly likely that users will create library files followed by updating their profile.*

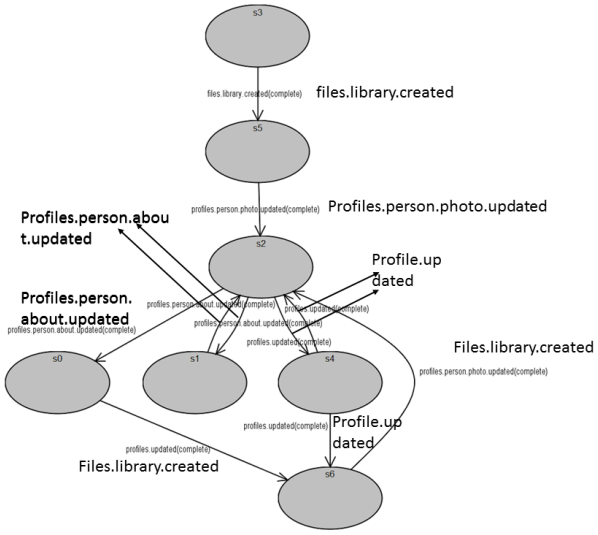


Figure 4.9: Best fitted process model with 0.0004 pruning ratio for model one (source:self-made)

4.4 Model two: predictive modelling impact on mainstream users

4.4.1 Introduction to the case

As discussed in the introduction of this chapter, the first three models have an extension to them i.e, focused on running the algorithm for the most frequent paths in the process graph. This model was also an extension to the first model and focused on finding the impact of predictive modelling on mainstream users of UniConnect in terms of RegPFA runtime and behaviour prediction. Mainstream users in this case are those users who perform same sequence of activities. To this end, the configuration was the same as the first model i.e., User_ID was taken as case_id, Event_name as activity_ID and Event_TS as the time stamp. Naturally, event log for this model also contained 241 activities, 1139 cases and 348,513 events as the configuration was similar to the first model. Since, this model focuses on predicting mainstream user behaviour filtering that users were the first step.

4.4.2 Log filtering

Mainstream users are those users who share some common sequence of activities. However, figure 4.10 shows that in unfiltered log most cases (around 90%) do not share the same sequences of activities that is most users perform a different set of events.

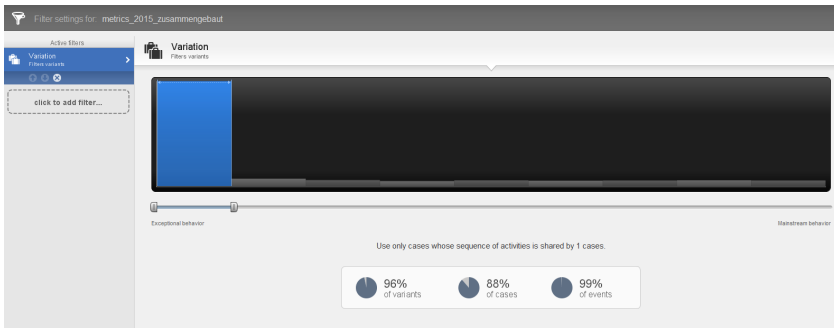


Figure 4.10: Exceptional behaviour in logs (source:self-made)

Therefore, to predict mainstream user behaviour, only cases demonstrating sequences of activities in common were considered. More specifically, only cases where a minimum of at least two users performed the same sequence of events as shown in figure 4.11

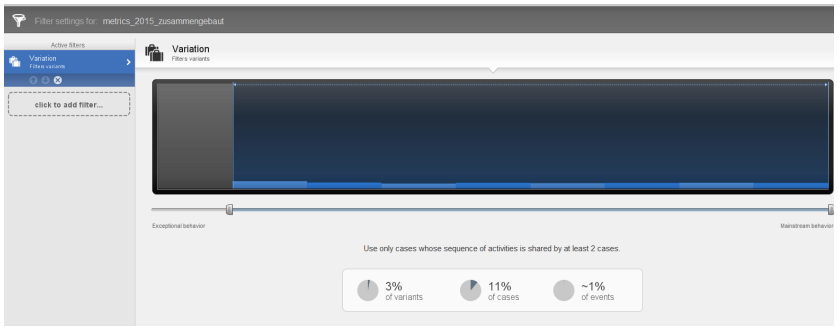


Figure 4.11: Mainstream behaviour in logs (source:self-made)

To filter the frequent paths Disco provides the variation filter which can filter the cases based on their sequence of activities. As it can be seen from the figure 4.11 there were only 11% of cases who shared common sequence of activities which leads to an exponential decrease in number of events as shown in table 4.10.

Statistics after variation filter	
Number of cases	129
Number of events	1881
Number of activities	2

Table 4.10: Statistics after the variation filter.

Since, the number of events and cases were reducing in a huge amount, information loss was certain. But, the objective of this model is to find the possibilities of mainstream behaviour in CSCW system or in other words to have a probabilistic process model for the users who have the same sequence of activities. Hence, information loss is tenable as this model does not concern to any other users except mainstream users. *However, there were only two activities left after filtering and one of them was empty and in large proportion. Therefore, this model is more about finding the effect of probabilistic modelling*

on mainstream behaviour in CSCW system (UniConnect) instead of user behaviour prediction (section 4.4.4). The next subsection discusses in detail the runtime and RegPFA parameters required to have an optimal process model for this configuration.

4.4.3 RegPFA mining & parameter modification

As discussed in the previous subsection, how filtering leads to tje most frequent variants for this particular model. This subsection explains how various iterations of RegPFA parameter modification (such as Maximum and Minimum states,EM iterations, etc.) leads to achieve a better fitted process model for the users who share the series of common sequence of activities.

First iteration of parameters that were used with the filtered logs are taken from the best fitted process model of model first as they have the similar configuration. Table 4.11 shows the parameters and their respective numbers.

RegPFA parameters	**statistics**
Minimum states	50
Maximum states	100
Regularization strength	0.2, 0.3
Number of tries	3
Model scorer	HIC
EM thrushold	0.001

Table 4.11: Parameter details for the first iteration of model two.

Runtime of the RegPFA with the parameters (4.11) was around four hours and according to the HEC model scorer, model with 57 states and a regularization strength of 0.3 was the best process model. However, EM was converging for the following models:

- In 213 EM iterations for the model having 74 states in it with a regularization strength of 0.2.

- Model having 75 states in it with a regularization strength of 0.2 and 0.3 in 270 and 293 EM iterations respectively.
- In 277 EM iterations for the model having 85 states in it with a regularization strength of 0.2.
- In 269 EM iterations for the model having 87 states in it with a regularization strength of 0.2
- Model having 88 states in it with a regularization strength of 0.2 and 0.3 in 246 and 235 EM iterations respectively
- In 269 EM iterations for the model having 95 states in it with a regularization strength of 0.3

The most important findings of this iteration was the *convergence of the EM was always in less than 300 iterations* and secondly, *process models with best model scorer was from 50 to 65 states*. Hence, in the second iteration for the search of the best fitted process model together with better runtime EM iterations was reduced to 300 and the number of maximum and minimum states was reduced from 50 and 100 to 20 and 65 respectively. Table 4.12 shows the RegPFA parameters for the second iteration. As expected, reduction in the number of states and lower EM iterations reduced the runtime to 2.5 hours and the best fitted process model had 36 states in it with a HIC score of 716.96. Whereas, the EM was converging after 192 iterations.

RegPFA parameters	**statistics**
Minimum states	20
Maximum states	65
Regularization strength	0.2, 0.3
Number of tries	2
Model Scorer	HIC
EM thrushold	0.001

EM iterations	300

Table 4.12: Parameter details for the second iteration of model two.

As the runtime was just 2.5 hours in this case so it can be said that if CSCW system has only those users who share the common sequence of activities, then the predictive process mining will be very feasible for the CSCW system. Moreover, the decision about the user events can easily be taken in real time.

Since, HIC was chosen as the model scorer its plot for all the process model between 20 and 65 states is shown in figure 4.12. It can be seen from this figure the HIC score was decreasing in the beginning but as the number of states was increasing HIC had some spikes. However, there was two sharp spikes at state 28 and 36 but the after state 36 there was sudden jump in HIC score which was the highest after any state. Moreover, there were some more fluctuations as well after state 36 but still there was an increasing trend. Hence, it can be argued that the best fitted process model had 36 states in it and if the states will increase its highly likely that HIC will increase with some small spikes as the gap between the downfall and rise of HIC value was highest two times between states 20 and 65. The next subsection applies the multiple pruning ratios to the best fitted process model achieved in order to predict the mainstream user behavior.

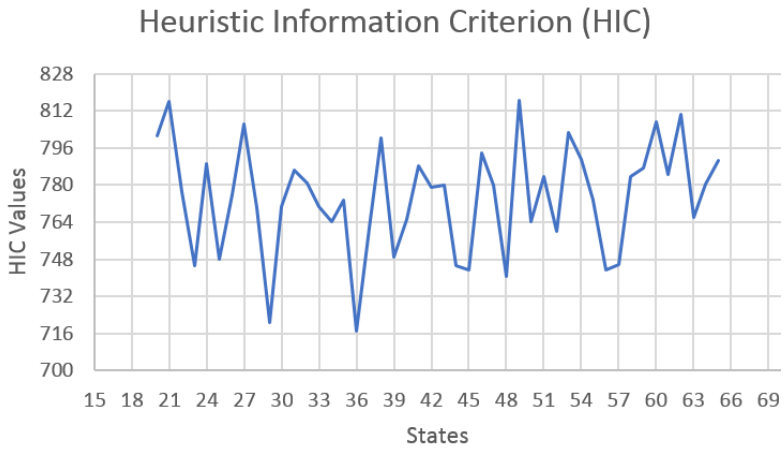


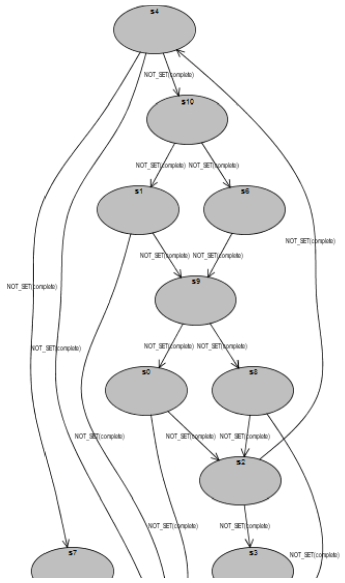
Figure 4.12: HIC for the best fitted process model of model 2 (source:self-made).

4.4.4 Mainstream user behaviour prediction

As discussed in the previous section, this model helped explore the possibilities of predictive process modelling in CSCW system (UniConnect), this section is focused more on predicting the user behaviour from the achieved best fitted probabilistic process model. However, the prediction of user behaviour here was not that important as there were only two activities and one of them was empty. While, on the other hand it was important to know what kind of pruning ratio would be optimal in case of mainstream behaviour, if there were more than two activities. Hence, to have a suitable pruning ratio it was important to analyze the best fitted model with different pruning ratios. Figure 4.13a, 4.14, 4.13b shows the probabilistic transition system with different pruning ratios and only the figure 4.14 was selected as the appropriate transition system because in figure 4.13a it was like spaghetti and not possible to see anything. However, in the figure 4.13b last one it was possible to analyze the transition system manually but there was significant information loss there was nothing to predict as the only activity that was left on the transition system was the empty activity (Not_Set).

The figure 4.14 was annotated to show the two activities as the picture from rapid miner was quite distorted. The two activities that were left pruning were the *empty activity (Not_Set)* and *file.library.created*. Hence, the only prediction that could be made was about the single activity (file.library.created) i.e., it was not really useful. On the other hand, the interesting point was when the pruning was applied a little less (0.07) than the optimal one (0.08) there was a total information loss. Hence, while deciding the pruning ratio a equilibrium between visibility and information loss is required in this kind of situations because for instance, if there are three activities and prediction about two of them is required it can be possible that one can be pruned completely if the pruning ratio is not suitable.

(a) Spaghetti process model with pruning ratio 0.4 (source:self-made).



(b) Unobservable process model with pruning ratio 0.09.

Figure 4.13: Latent visualizations for model two (source:self-made).

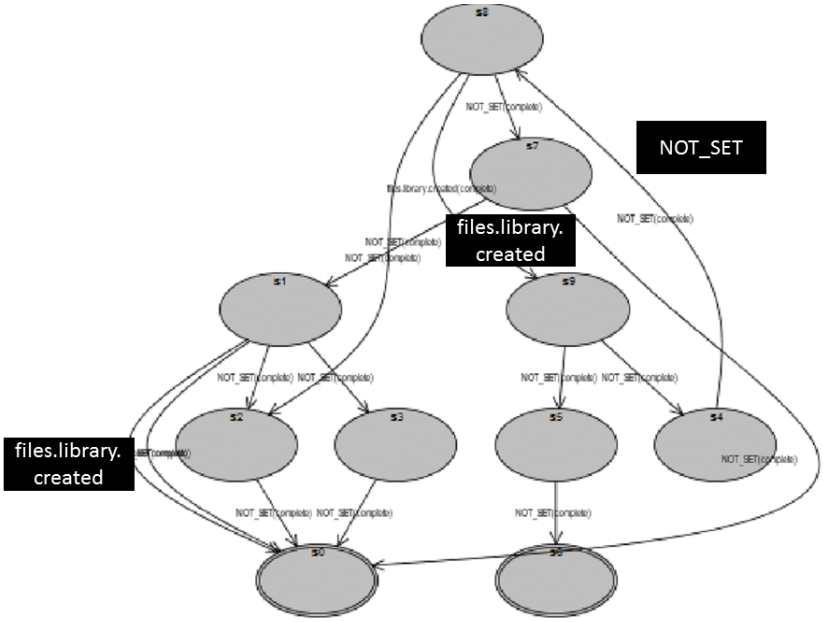


Figure 4.14: Optimal process model with pruning ratio 0.08 for model two (source:self-made).

4.5 Model three: prediction of communities that users will join.

4.5.1 Introduction to the case

This model is focused on predicting the communities that the users will join while being a part of some other community. Moreover, it also states which communities are highly likely to have same users. In order to have the correct predictions selection of relevant columns from the dataset were required. Appropriate selection of data columns helped in achieving a significant configuration for the input of the RegPFA algorithm. To this end, User_ID was taken as case_id, COMMUNITY_ID as activity_id, ITEM_UUID as otherColum_id and EventTS as the time stamp. Eventlog for this model have 348,513 events which were classified in 412 activities. The 1107 variants were executed in 1139 cases. After inspecting the logs briefly in Disco it was founded that *relative frequency of*

35 activities out of 412 was 90%, making it an appropriate sample for the analysis. In Terms of UniConnect it meant that majority of the user belong to these 35 communities as shown in figure 4.15.

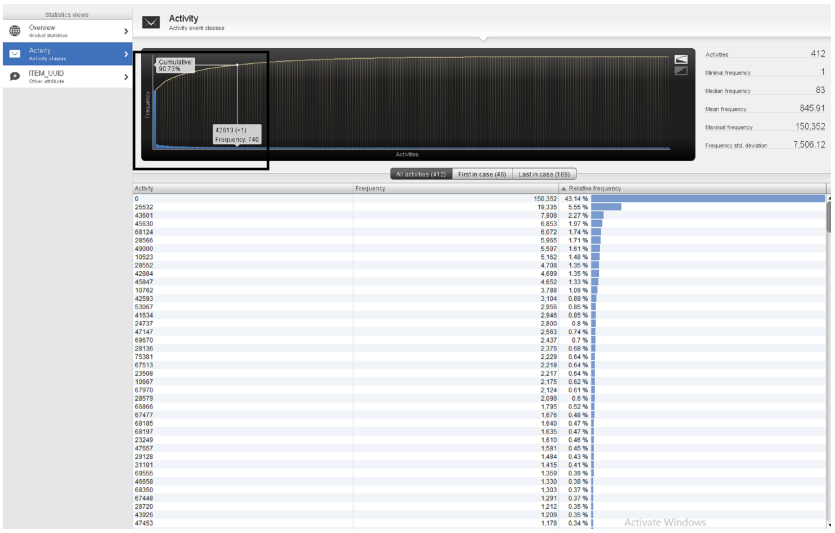


Figure 4.15: High cumulative frequency of few activities indicated by a black rectangle (source:self-made).

As it can be clearly seen from the graph marked by the rectangle in the figure 4.15 that cumulative frequency of only a few activities were around 90%. This brief view on log file in Disco shows the intricacy of dataset while defining the log filtering certainty.

4.5.2 Log filtering

To filter the exceptional behaviour mentioned in previous subsection, Disco attribute filter was used. It filters out on the basis of some particular cases and activities. Table 4.13 shows the statistics before and after the filtration process.

Statistics before and after filtration of communities		
Number of cases	1139	162
Number of events	348513	109345
Number of activities	412	35

Mean case duration	11 weeks	31 weeks
Median case duration	17 weeks	32 weeks

Table 4.13: Statistics before and after applying the attribute filter in model three.

As can be seen from the table 4.13, there was a huge drop in the number of activities, cases and events hence, leading to a lot of information loss. However, this information was not important for the analysis as this information constitute only ten percent in the analysis or in other words useless information. Thus, it can be argued that information loss was critical in some situations as it in was in model one. Moreover, table 4.13 shows that before filtering the useless information the gap between the mean and median case duration was six weeks, indicating the presence of outliers. However, after filtering it was reduced to one week which was acceptable as the some of the cases were running for even one year. Next subsection discusses how modifying the parameters in different iterations of RegPFA helps achieve the suitable probabilistic process model.

4.5.3 RegPFA mining & parameter modification

As the configuration (case_id, activity_id) and respective counts for this model is different as compared to the mode one, exploration of parameters should be done in various iterations for the best fitted process model with admissible runtime. First iteration of parameters that were used with the filtered logs (table 4.13) is shown in table 4.14. The filtered logs contained 14.2% of cases, 8.4% of activities and 31.3% of events in comparison to the 100% of cases, activities and events that were used in naive run of RegPFA (section 4.1).

RegPFA parameters	**statistics**
Minimum states	100
Maximum states	150
Regularization strength	0.2, 0.3,0.5

Number of tries	2
Model scorer	HIC
EM threshold	0.001
EM iterations	500

Table 4.14: Parameter details for the first iteration of model three.

The runtime of RegPFA algorithm was a never-ending process for this iteration. There were only ten process models after 13 days between the minimum and maximum states of 100 and 150 respectively. Moreover, there was no model where EM was converging even after a low EM threshold of 0.001 and 500 EM iterations. Hence, to improve the grid search in next iteration number of minimum and maximum states were reduced to 45 and 110 respectively while keeping the other parameters same.

Table 4.15 shows the parameter used for second iteration. Reducing the limit of minimum and maximum number of states leads in completion of RegpFA algorithm with a runtime of around two weeks. However, to have a best fitted model EM must converge for at least some process models between the minimum and maximum states; if not for the process model with the lowest HIC score. The process model that has lowest HIC score had 72 states in it.

RegPFA parameters	**statistics**
Minimum states	45
Maximum states	110
Regularization strength	0.2, 0.3,0.5
Number of tries	2
Model scorer	HIC
EM threshold	0.001
EM iterations	500

Table 4.15: Parameter details for the second iteration of model three.

To overcome the problem of EM converging in second iteration this iteration decrease the threshold of EM convergence from 0.001 to 0.0001 in search of best fitted process model. This decrease will definitely affect the runtime as EM will take more time to converge. However, it was noticed in the previous run that the best fitted process model had a regularization strength of 0.3. Hence, to have a best fitted model with a suitable runtime only regularization strength of 0.3 was used. Table 4.16 shows the parameters used for this iteration.

RegPFA parameters	**statistics**
Minimum states	45
Maximum states	110
Regularization strength	0.3
Number of tries	2
Model scorer	HIC
EM threshold	0.0001
EM iterations	500

Table 4.16: Parameter details for the third iteration of model three.

After running for nine days and three hours the RegPFA algorithm was finally over and the best fitted process model had 77 states in it with HIC score of 74720.21. However, EM was converging here after 476 iterations only for the process model that had 67 states in it. Hence, decreasing the EM threshold helped achieve the better fitted process model. One interesting point to observe in this iteration was the single digit runtime. In the previous iteration it was around two weeks and in this iteration also it could have gone easily to more than two weeks but, since only a regularization strength of 0.2 was used it remained in single digit. Thus, *a balance between runtime and parameters was important to have a best fitted model in suitable runtime.*

Figure 4.16 shows the HIC plot for the best fitted parameters (table 4.16) of RegPFA. The plot shows the HIC value on the y-axis compared to the number of states of a mined

model on the x-axis. It can be seen from the figure there were spikes for almost every five states but the biggest spike where the the gap between the downfall and rise of HIC value came out highest, was at state 77 with a HIC score of 74720.21. Moreover, after state 77 HIC value showed an increasing trend with some spikes. Hence, it can be said if the number of states will increase its highly likely that HIC will show an increasing trend i.e, not falling again to 74720.21 and that makes the process model with 77 states as the best fitted process model. The next subsection applies the multiple pruning ratios to the best fitted process model achieved in order to predict the communities that user will join.

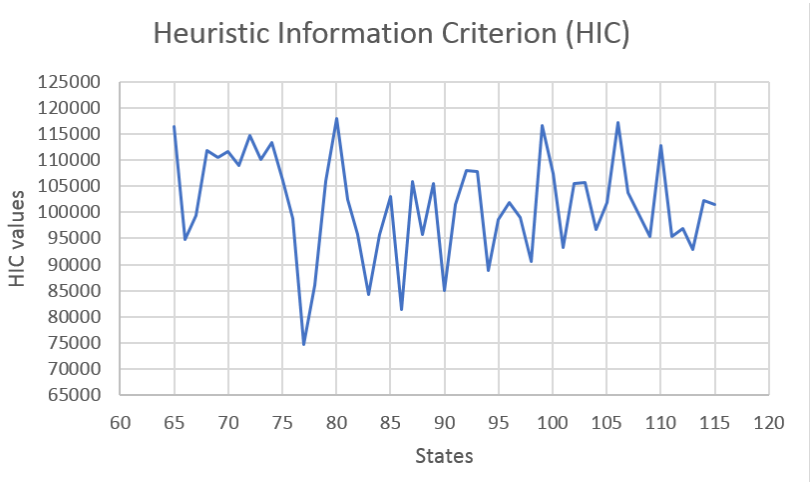
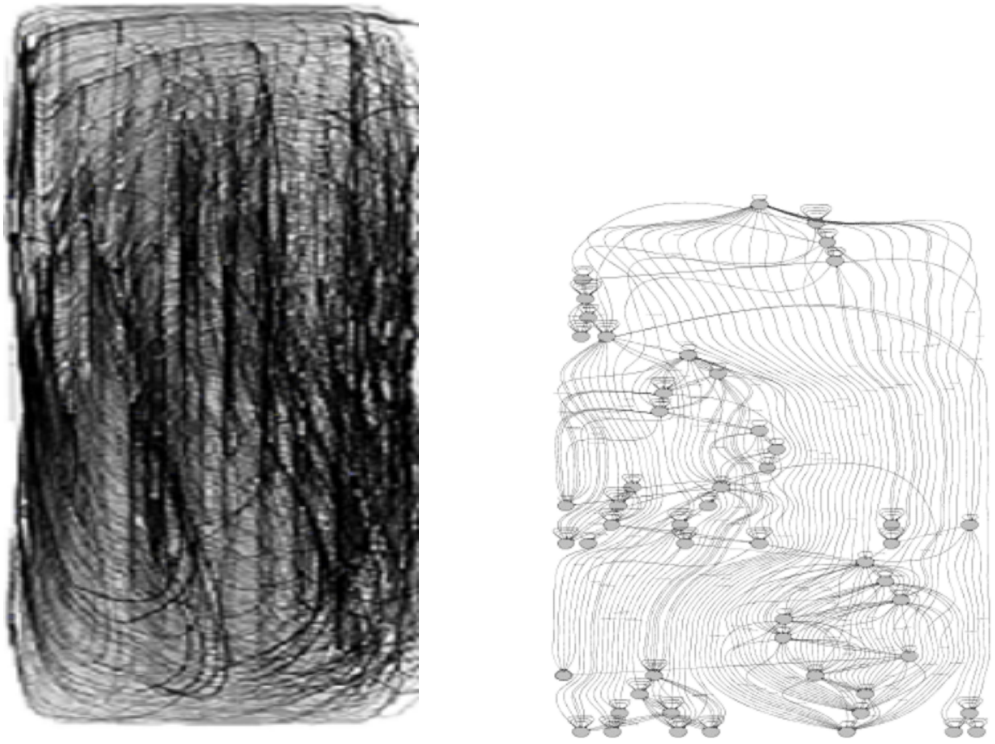


Figure 4.16: HIC plot for the best fitted process model of model three (source: Self-made).

4.5.4 Community joining predictions

To predict the communities that will be joined by the user in near future, analysis of probabilistic transition system produced was critical. Evidently the transition system (process models) even with very low pruning ratio was like a spaghetti process model because of the huge number of events and community joining predictions from them also proved insurmountable. Figure 4.17a shows the spaghetti transition system created by rapid miner with a very low pruning ratios of 0.25. To have a process model that can be manually visualized it was important to decrease the pruning ratio hence, for that pruning ratio

was decreased to 0.15. As it can be seen from figure 4.17b that transition system was recognizable than the spaghetti one but, still can not be analyzed manually.



(a) Insignificant process model with pruning ratio 0.4.

(b) Latent process model with 0.07 pruning ratio.

Figure 4.17: camouflaged visualization for model three (source:self-made).

Hence, for the manual analysis of transition system produced by Rapid miner pruning ratio was further reduced from 0.15 to 0.1 in .2 steps (i.e, to .13 and .1) and only at 0.1 it was manually possible to predict the communities that will be joined by the user. Figure 4.18 shows the corresponding process model (transition system) with 0.1 pruning ratio. As can be seen from figure 4.18, for states So and S1 there was a path to reach every other state that is why they paths are highlighted with red and black rectangles. The number on the edges shows the COMMUNITIES_ID (activity_id). Moreover, after pruning there were only four activities left out of 32; hence, the predictions can be done only about these communities by following the paths from one state to another in transition diagram. The

predictions that can be done about the communities of UniConnect that users will join are as follows:

- *Users of community zero will most likely to join 68197 and 68124 communities and vice versa.*
- *Users of community 68197 will most likely to join 68124 and vice versa.*
- *As there were some self loops present in the process models (S0, S1, S2) so its highly likely that users of these communities i.e., zero, 68197, 68555 and 68124 would be the same.*

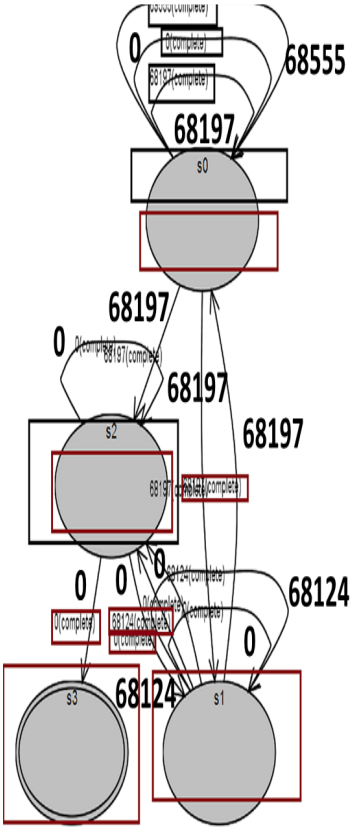


Figure 4.18: Best fitted process model with Pruning ratio 0.1 for model 3 (source:self-made).

4.6 Model four: predictive modelling for mainstream user behaviour

4.6.1 Introduction to the case

This model is an extension of model three and like model two it also focused on finding the impact of predictive modelling on mainstream users of UniConnect in terms of RegPFA runtime and behaviour prediction. Mainstream users in this case are those users who share the same communities. To attain this goal it was important to select only that communities that were followed by more than one user or in other words cases that shared the common sequence of activities. Hence, purposefully USER_ID was taken as case_id, COMMUNITY_ID was taken as activity_id and ITEM_UUID was taken as otherColumn_ID. Before filtering this model had 348,513 events which were classified in 412 activities and 1139 cases which was similar to previous model (model 3) as the configuration was similar. Since, the cases that were needed was the only ones who shared the common sequence of activities. Thus, filtration of that cases should precede before digging into the process of RegPFA mining and user behaviour prediction.

4.6.2 Log filtering

Users that belonged to common communities were only a few. In other words it can also be said that there are a few users who follow the same path in the process map as it can be seen from figure 4.19 that there were only three percent of cases who follow the same path and had activities in common .

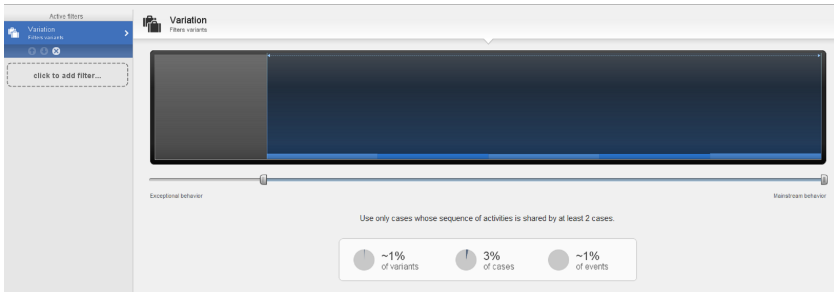


Figure 4.19: Disco showing cases that share common sequence of activities.

Therefore, filtration of these users were required in order to predict the common communities followed by the users. To filter the frequent paths (mainstream behaviour) variation filter of Disco was used. Table 4.17 shows the statistics before and after applying the variation filter.

Statistics before and after the variation filter		
Number of cases	1139	42
Number of events	348513	215
Number of activities	412	2

Table 4.17: Statistics before and after applying the attribute filter in model four.

As it can be seen clearly from the table that the number of events and cases were reducing in a huge amount thus,information loss was certain. But, the main objective of this model is to explore the prediction possibilities of mainstream user behaviour in CSCW system or in other words to have a probabilistic process model for the users who share the common sequence of activities. Hence, the information loss is justifiable as this model does not concern to any other users except the mainstream users. The next subsection discusses in detail the runtime and RegPFA parameters required to have an optimal process model for this configuration.

4.6.3 RegPFA mining & parameter modification

This subsection discusses how RegPFA mining and parameter modification (such as Maximum and Minimum states,EM iterations, etc.) helps in achieving a better fitted process model for the users that belong to more than one community.

First iteration of parameters were chosen from the best fitted process model of model second as model second also runs the algorithm for most frequent paths and had same number of activities. Table 4.18 shows the parameters and their respective numbers used for this model. The filtered logs contain 73.5% of cases ,97.08% of activities and 15% of events.

RegPFA parameters	**statistics**
Minimum States	20
Maximum States	65
Regularization strength	0.2, 0.3
Number of tries	2
Model Scorer	HIC
EM thrushold	0.001
EM iterations	300

Table 4.18: Parameter details for the first iteration of fourth model.

As expected, the best fitted process model was discovered after one hour in second try with a HIC score of 286.4. Discovered process model had 27 states in it with EM converging after 234 iterations. *However, an interesting founding of this model was that the number of events and cases were little high in model second but, still the RegPFA parameters of model two were optimal to discover a good fitted process model.* Moreover, as the runtime was just one hour it shows the feasibility of predictive process discovery in the CSCW system where only that cases are considered who share the common sequence of activities.

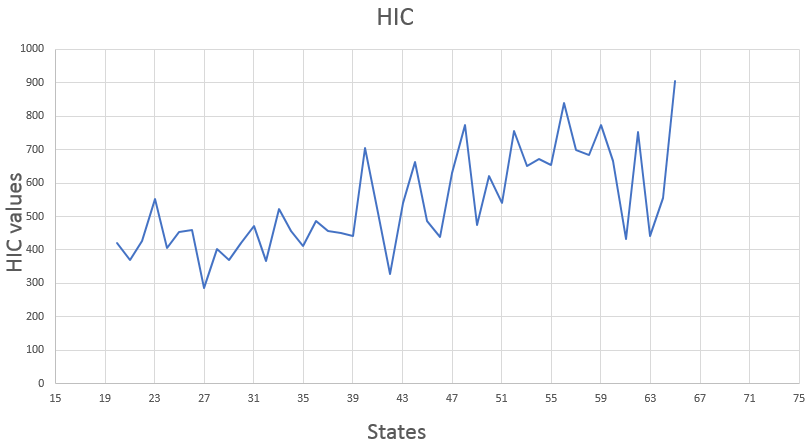


Figure 4.20: HIC for the best fitted process model of model four (source:self-made).

Since, HIC was chosen as the model scorer its plot for all the process model between 20 and 65 states is shown in figure 4.20. It can be seen from the figure the HIC values showed an increasing trend between minimum and maximum state. There were several sharp spikes in the plot but the lowest value of HIC was attained quite early in the plot i.e., at state 27. However, HIC was maximum at state 65 while showing some deep downfall at state 63 and 60 but the overall trend was ascending hence it can be argued if the number of states will increase HIC will increase as well.

4.6.4 User behaviour prediction

As indicated in previous subsections the main objective of this model was to explore the possibilities of probabilistic modelling for the mainstream users in UniConnect rather than their prediction. As after filtering there were only two activities left and to have predictions for two activities is not really useful. In spite of having different number of events, cases and activities than model two the results for the RegPFA parameters was quite similar to model two. Moreover, faster runtime also indicates the effectiveness of predictive modeling when used for a certain user group in CSCW system.

4.7 Model five: predicting activities performed by users in communities

4.7.1 Introduction to the case

To fully explore the possibilities of probabilistic modelling in CSCW system it was important to endeavour all the configurations possible. Thus, this model combine two columns as activity_id i.e., USER_ID and EVENT_NAME while taking COMMUNITY_ID as case_id and EVENT_TS as timestamp. Event log for this model had 348,513 events which were classified in 13,288 activities and 412 cases. A brief inspection of event logs illustrates the two very exceptional characteristics of the log file that are given below:

- *Majority of the activities were empty from EVENT_NAME column*
- *There were a majority of activities from USER_ID column that were singleton i.e., not in combination with the EVENT_NAME column as shown in figure 4.21. This behaviour is either because of the empty activities or because of the different activities from EVENT_NAME column present in disparate communities (case_id). Moreover, the relative frequency of this actvities was quite high.*

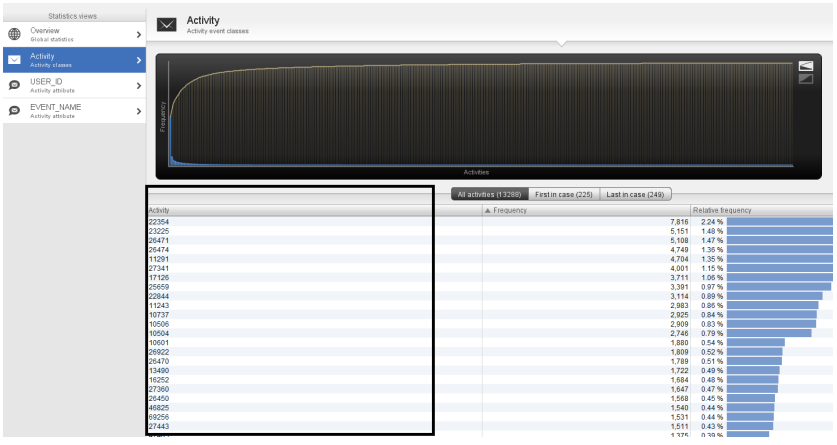


Figure 4.21: Singleton activities shown by Disco (source:self-made).

This exceptional behaviour calls for the need of filtering in order to discover the optimal

process model.

4.7.2 Log filtering

To filter the unusual behaviour i.e., to remove the empty activities and the singleton activities Disco attribute filter was used. Unlike model first, in this model the order of removing the empty and the singleton activities was not important because the activities that were needed in the analysis should be a combination of USER_ID and EVENT_NAME in order to predict which user will perform certain events in the various communities. Table 4.19 shows the statistics before and after removing the empty and Singleton activities.

Statistics before and after empty and singleton activities filtered		
Number of cases	412	253
Number of events	348513	1314
Number of activities	13288	12

Table 4.19: Statistics before and after applying the attribute filter in model five.

As it can bee seen from the table there were only 12 activities left after filtering the empty and singleton activities while having 1314 events classified in 253 cases. However, this huge loss in number of events and activities can not be considered as information loss as this model focuses only on that activities where there is an user (USER_ID) performing an event (EVENT_NAME) in any community. The next subsection discusses in detail the runtime and RegPFA parameters required to have an optimal process model for this configuration.

4.7.3 RegPFA mining & parameter modification

In order to achieve a better fitted process model, for the communities where users perform a certain event, this subsection discusses various iterations of RegPFA mining and parameter modification(such as Maximum and Minimum states, EM iterations, etc.).
Similar to previous model, first iteration of parameters for this model were also chosen

from the best fitted process model of model two they both have approximately same number of cases, activities and events. The filtered logs contain 61% of cases, .1% of activities and 1% of events. Table 4.20 shows the parameters and their respective numbers used for this model.

RegPFA parameters	statistics
Minimum states	20
Maximum states	65
Regularization strength	0.2, 0.3
Number of tries	2
Model scorer	HIC
EM thrushold	0.001
EM iterations	300

Table 4.20: Parameter details for the first iteration of model five.

Runtime of this iteration was around four hours and according to HEC (KPI), model having 34 states in it was the best fitted process model with EM converging after 234 iterations. However, the model was quite optimal and good fitted but to check the effect of EM iterations and threshold on EM convergence for the best fitted process model it was decreased and increased respectively in next iterations.

In second iteration to find a better fitted process model, EM iterations and threshold was decreased and increased respectively while keeping the other parameters similar to first iteration. Table 4.21 shows the parameters used for this iterations.

RegPFA parameters	statistics
Minimum States	20
Maximum States	65
Regularization strength	0.2, 0.3
Number of tries	2
Model Scorer	HIC

EM thrushold	0.001
EM iterations	300

Table 4.21: Parameter details for the second iteration of model five.

When the EM iterations was decreased from 300 to 150 and the EM threshold was increased to 0.01, EM was converging for the model having 24 states in it after 69 iterations. Moreover, it also produced the lowest HEC score i.e., 747.21. Hence, the best fitted process model had 24 states in it with EM converging after 69 iterations while having the EM threshold and the HEC score of 0.01 and 747.21 respectively. The model had the runtime of three and half hours

HIC plot of best fitted process model is shown in figure 4.22. The plot shows the HIC value on the y-axis compared to the number of states of a mined model on the x-axis. It can be clearly seen from the figure that HIC was showing an increasing trend and the biggest drop was in the beginning at state 27 to produce a HIC score of 747.21. However, as the trend was strictly increasing after state 27 it can be stated that HIC will increase with the increase in number of states while producing the best fitted process model at state 27.

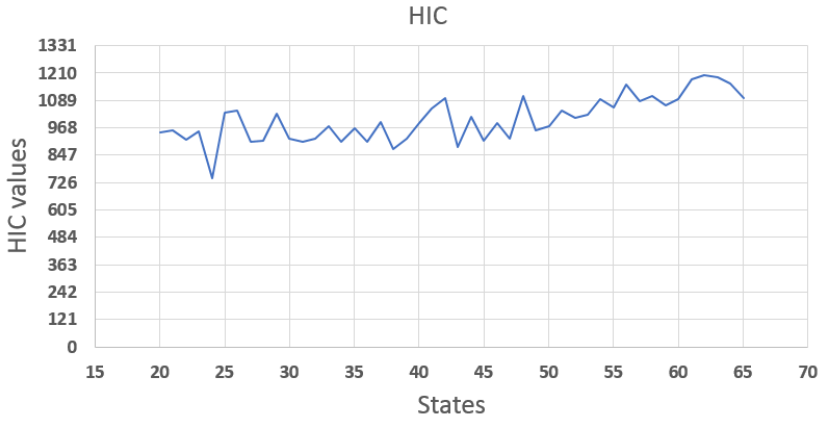


Figure 4.22: HIC plot for model five (source:self-made).

4.7.4 User behaviour prediction

Even though the number of activities was not that high, the visualization of transition systems at very low pruning ratio proved invincible. As it can be seen from the figure 4.23 that the pruning ratio was quite low i.e., 0.03, still it was impossible to analyze the process model manually. Moreover, before this pruning ratio i.e. from 0.1 to 0.03 there were only spaghetti process model produced. Thus, the pruning ratio was lowered to 0.015 because the process models that were produced from pruning ratio 0.03 to 0.015 were not even mildly visible. Figure 4.24 shows the process model with pruning ratio 0.015 that was moderately visible but, still difficult to analyze manually.
Hence, to predict the activities performed by users in communities pruning ratio was lowered further to 0.1 as shown in figure 4.25. As the pruning ratio was quite low, there were only three activities out of 12 were left. To have a better visualization activities are shown with the help of coloured rectangles as the picture from Rapid miner was quite distorted. *1) The red coloured rectangle indicates the activity* ***file collection was updated by the user with id 2.*** *2) The brown rectangle shows the activity* ***file collection was created by the user with id 2.*** *3) The yellow rectangle shows the activity* ***forum was deleted by the user with id 10060.*** Starting from the state S0, the two major predictions that can be made highly likely about this model are as follows:

- User 2 will create the file collection followed by updating the file collection i.e, from state S0 to S7 and then to S14 as indicated by black rectangles. As this pattern was quite repetitive in some other states as well so they are shown as well with the help of black rectangles for instance, S0 to S11 and then to S3

- User 2 will create the file system followed by the deletion of forum by user 10060

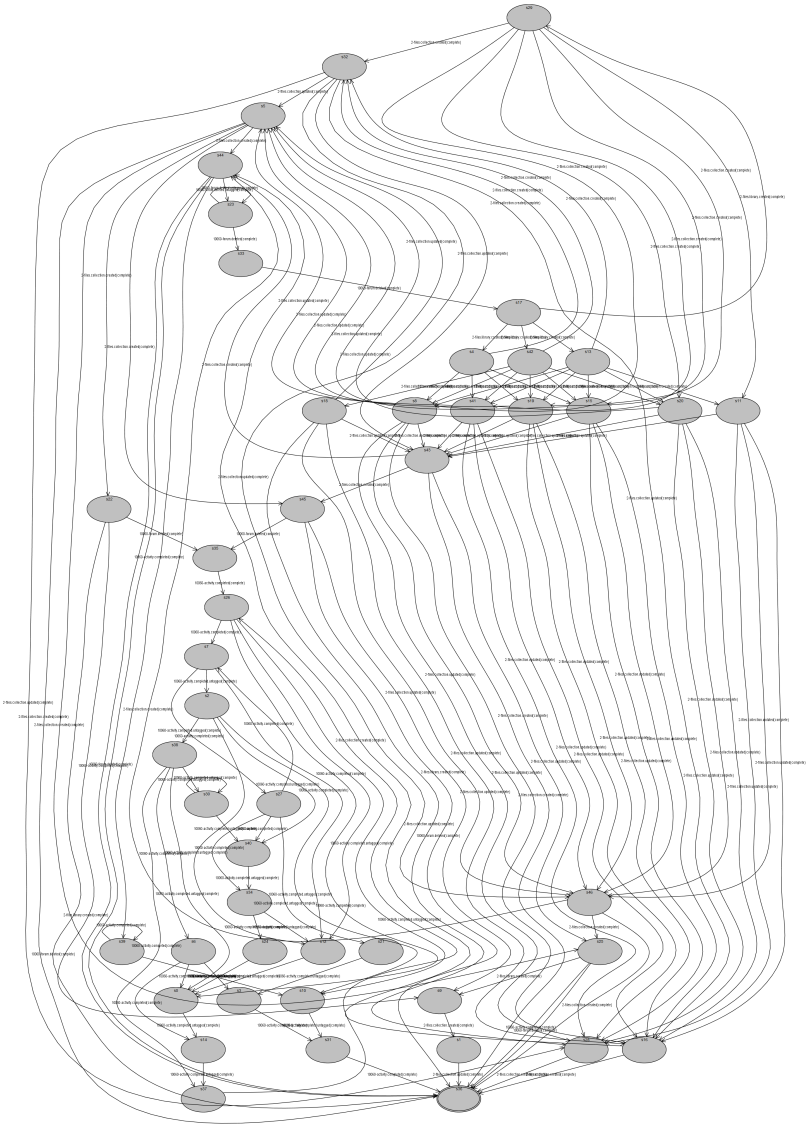


Figure 4.23: Transition state diagram with pruning ratio 0.03 for model five (source:self-made).

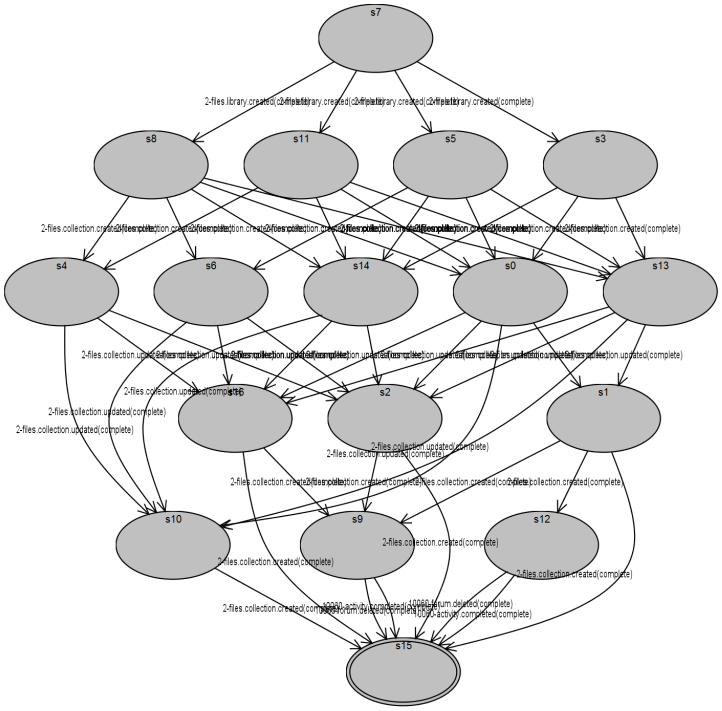


Figure 4.24: Transition state diagram with pruning ratio 0.015 for model five (source:self-made).

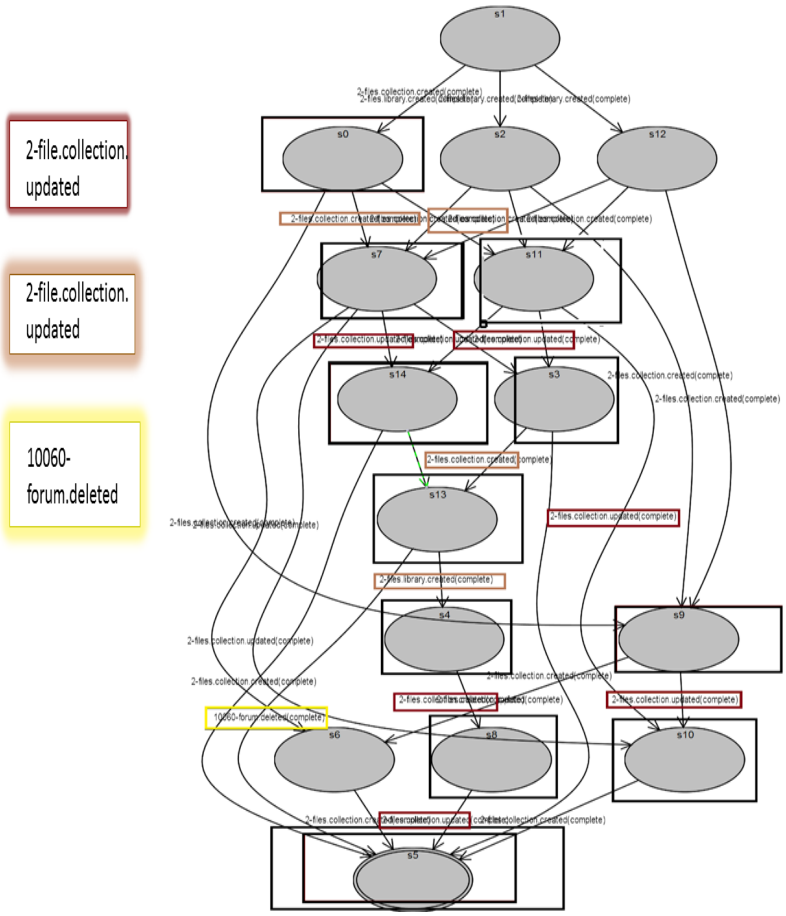


Figure 4.25: Best fitted process model with pruning ratio 0.01 for model 5 (source:self-made).

4.8 Model six: predicting events that will happen inside communities

4.8.1 Introduction to the case

The focal point of this model was to predict the events that takes place inside the communities of UniConnect platform. Correct prediction of events required the selection of appropriate columns from the dataset and in order to fulfill that COMMUNITY_ID was taken as case_id, EVENT_NAME was taken as activity_id and EVENT_TS as the times-

tamp. Event log for this model had 348,513 events which were classified in 241 activities. The 378 variants were executed in 412 cases.

Even after the deep inspection of log file there was only one exceptional behaviour found that is *the relative frequency of empty activities was 77.9%* as shown in figure 4.26. In terms of UniConnect it means that events that took place inside communities were not equally spread and removal of this empty events were necessary to have an optimal process model. Hence, log filtering was required before the mining and prediction process.

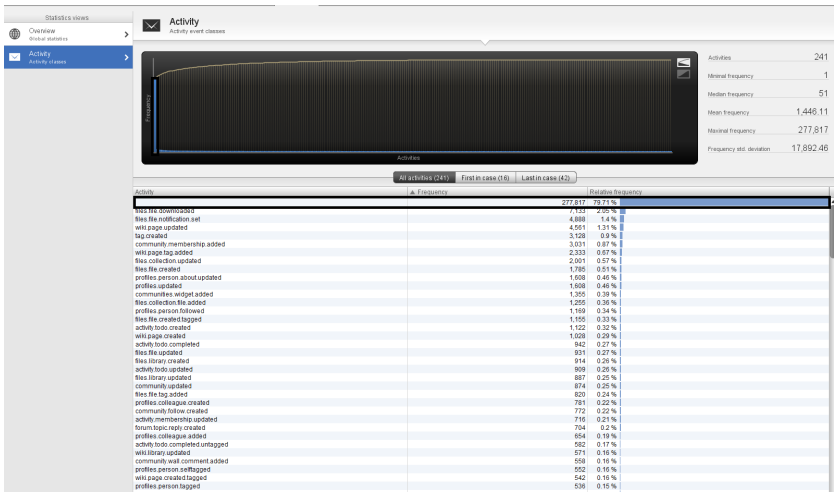


Figure 4.26: Disco showing high relative frequency of empty activities (source:self-made).

4.8.2 Log filtering

As the only unusual behaviour found was the empty activities and its higher relative frequency, hence they must be filtered. To filter the empty activities out, Disco attribute filter was used as it can filter on the basis of cases and activities. Table 4.22 shows the before and after filtration statistics of the configuration.

Statistics before & after attribute filter		
Number of cases	412	362
Number of events	348513	1314

Number of activities	241	240

Table 4.22: Statistics before and after applying the attribute filter in model six.

As it can be seen from table 4.22 there was a huge drop in number of events but reduction in number of cases was very low which indicates close to zero information loss. Moreover, the information that was filtered out in terms of number of cases was not that important anyhow for the scenario.

4.8.3 RegPFA mining & parameter modification

As discussed in the previous subsection how filtering played a vital role in log quality of this particular model, this subsection explains how various iterations of RegPFA parameter modification (such as Maximum and Minimum states,EM iterations, etc.) together with filtering not only leads to achieve a better runtime of algorithm but also a better fitted process model.

First iteration of parameters were chosen by taking into consideration the best fitted process model of model three as its number of events and cases were in the range of this model. The filtered logs contain 87% of cases, 99% of activities and 20% of events as compared to unfiltered logs. Table 4.23 shows the parameters of RegPFA used for this iteration.

RegPFA parameters	**statistics**
Minimum States	45
Maximum States	110
Regularization strength	0.2, 0.3
Number of tries	2
Improvement threshold	3
EM threshold	0.0001
EM iterations	500

Model Scorer	HIC

Table 4.23: RegPFA parameter for the first iteration of model six.

After running for two Weeks and 11 hours the RegPFA algorithm was finally ending and the best fitted process model had 62 states in it with HIC score of 74720.21. However, EM was never converging for any process model between minimum and maximum states. Hence, there was the need of increasing the number of EM iterations in the next iteration of parameter modification and RegPFA mining.

As mentioned in previous iteration there was a need of increasing EM iterations, this iteration increase the EM iteration from 500 to 700 for the EM convergence. However, this increase in runtime will definitely increase the runtime as EM for every process model will run more than 500 times. RegPFA parameters used for this iteration are shown in table 4.24.

RegPFA parameters	**statistics**
Minimum States	45
Maximum States	110
Regularization strength	0.2, 0.3
Number of tries	2
Model Scorer	HIC
EM threshold	0.0001
EM iterations	700

Table 4.24: Parameter details for the second iteration of model six.

As expected, runtime (i.e., two weeks, three days and 15 hours) was increasing by three days as compared to previous iteration and EM was converging for some process models between the minimum and maximum states. *However, the best fitted process model had 92 states in it with a regularization strength and HEC score of 0.2 and 99957.69 respectively.*

EM was also converging after running 653 times in first try. Hence, increase in EM iterations facilitated in discovering a better fitted process model.

Since, HIC was chosen as the model scorer its plot for all the process model between 45 and 110 states is shown in figure 4.27. It can be seen from the figure the HIC values were fluctuating continuously while having three sharp downward spikes at states 72, 86 and 92. However, the HIC score at state 86 and 92 very pretty close but the process model that had 86 states in it came out with lowest HIC score while after state 92 HIC showed an strictly increasing trend. Therefore, it can be argued that if the number of states will increase HCI will keep on rising as well. Hence, process model that had 86 states in it were chosen as the best process model with the parameters mentioned in table . Next subsection predicts the events that will take place inside the communities in near future by applying different pruning ratio to discovered process model.

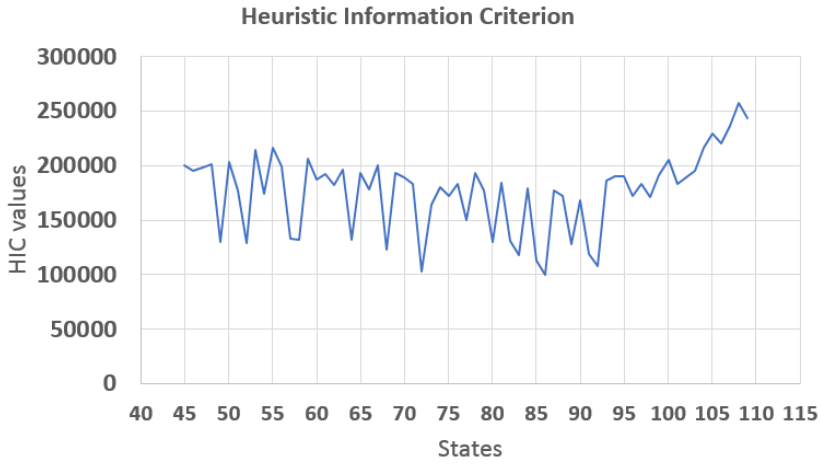


Figure 4.27: HIC plot for best fitted model of model six.

4.8.4 Activities prediction

To predict the activities that would be performed inside the communities in near future, analysis of discovered probabilistic transition system was required with different pruning ratios. On the other hand, the discovered transition system even with very low pruning ratios was tangled like a yarn and not very easy to comprehend manually. Whereas, activ-

ity prediction was the manual process so accomplishing it manually from the ascertained transition systems proved insurmountable even after very low pruning ratios. Figure 4.28 and 4.29 shows the discovered process model with 0.1 and 0.04 pruning ratios. Both the figures were very cluttered and it was impossible to analyze them manually.

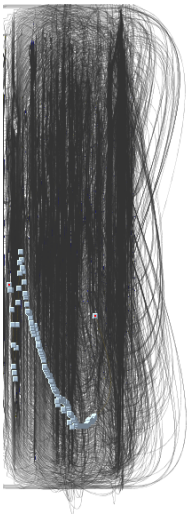


Figure 4.28: Unobservable process model of model six (source:self-made).

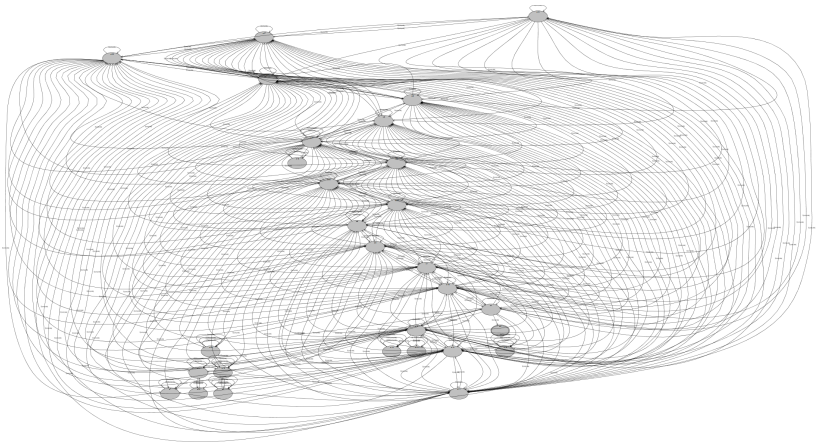


Figure 4.29: Transition state diagram with 0.04 pruning ratio (source:self-made).

Hence, lowering of pruning ratios were required as pruning ratio determines how many edges from the transition system should be kept in visualization. Lowering of pruning

ratio should be done by maintaining a balance between information loss and comprehensibility. Thus, to attain that pruning ratio was reduced continuously until 0.0004 because all the process models between 0.04 to 0.0004 could not be visualized and analyzed manually. Figure 4.30 shows the process model for 0.004 pruning ratio.

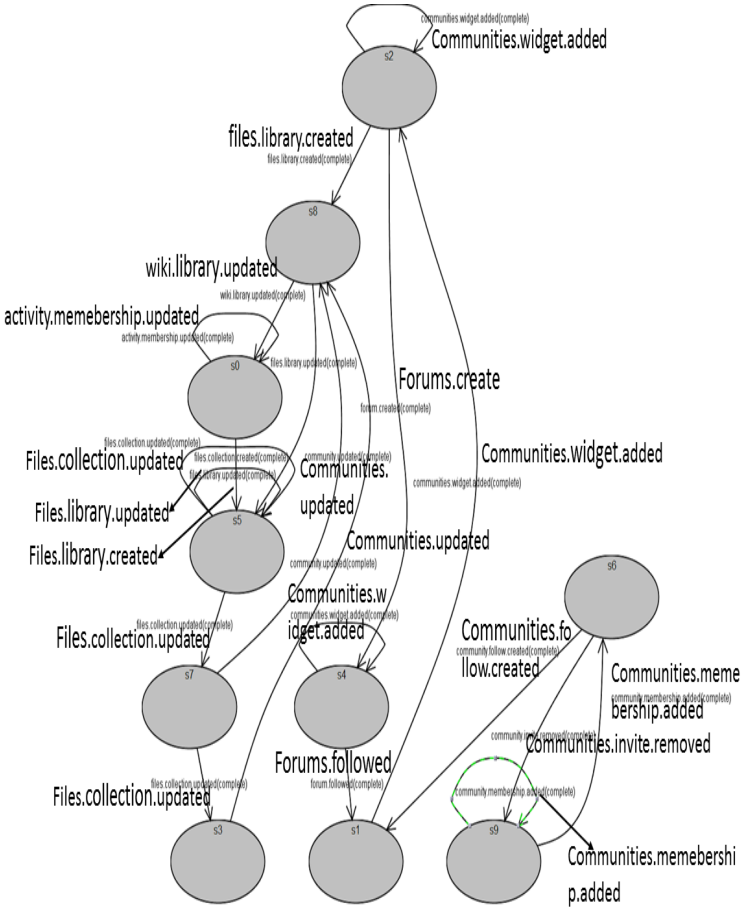


Figure 4.30: Comprehensive process model for model 6 with 0.0004 pruning ratio (source:self-made).

As can be seen from figure 4.30 after pruning there were only 12 activities left out of 240 and the paths to go from one state to another was also not that high. Even though the paths and activities were not that high, still it was not manually possible to cover every path from one state to another. That is why predictions were limited to two steps i.e., one activity followed by another activity. Starting from state S0 to S9 all the highly likely

events that would take place inside communities in two steps are shown in figure 4.31. All the predictions were separated by a blue line to have a clear view. Another reason to show the predictions diagrammatically was that meaning of some activities was not known hence could not be described. It would have been possible to prune the transition system even more but the pruning ratio was already too low, *if reduced further could add pseudo observations i.e., underfitted model.*

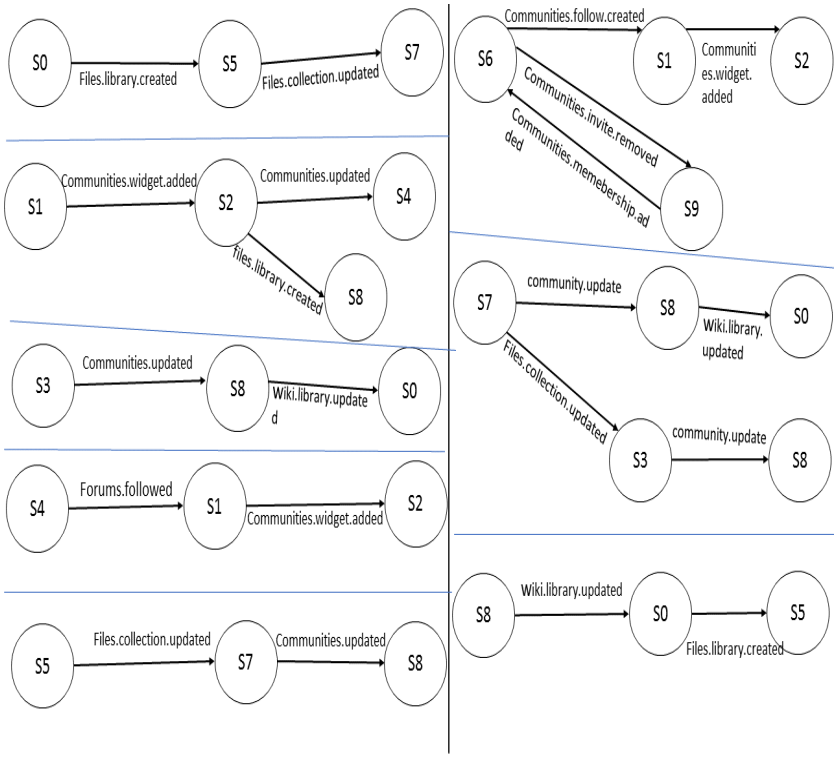


Figure 4.31: Predictions for model six (source:self-made).

4.9 Models findings

This section provides findings of the all the models in terms of input configurations (such as, number of events) and RegPFA parameters that can be used later while applying probabilistic model (RegPFA) to some other CSCW system or to UniConnect. Runtime for all the models were around four to five months in order to explore all the possible configura-

tion for UniConnect.

- When the number of events are in the range of 25000 to 50000 and are of different types i.e., having many activities (above 200) then the RegPFA parameters that can be optimal are given in table 4.25.

RegPFA parameters	**statistics**
Minimum states	50
Maximum states	100
Regularization strength	0.2,0.3
Number of tries	2
Model scorer	HIC
EM thrushold	0.001

Table 4.25: Parameter details for the number of events between 50000 and 75000

- When the number of events are in the range of 100000 but the event type is same i.e, having few activities (less than 50) then the choice that would be optimal for RegPFA parameters are given in table 4.26

RegPFA parameters	**statistics**
Minimum States	45
Maximum States	110
Regularization strength	0.3, 0.5
Number of tries	2
Model Scorer	HIC
EM threshold	0.0001
EM iterations	500

Table 4.26: Parameter details for the number of events around 100000.

- When the number of events are in the range of 50000 to 75000 and having many different types of activities then the optimal choice of RegPFA parameters is given in table 4.27

RegPFA parameters	**statistics**
Minimum States	45
Maximum States	110
Regularization strength	0.2, 0.3
Number of tries	2
Model Scorer	HIC
EM threshold	0.0001
EM iterations	700

Table 4.27: Parameter details for the number of events between 50000 and 75000.

- Lastly, when the number of events are quite low (less than 2000) and had few activities (less than 20) then the different parameters have to checked as in this research they were quite fluctuation (Model two, four and five) in terms of EM iterations and threshold. For example, in model two and four where the number of events were 1881 and 215 while having two activities each, the EM threshold and iteration was 0.001 and 300 respectively. Whereas, in model five where the number of events and activities were 1314 and 12, the optimal EM threshold and iterations were 100 and 0.01 respectively. That is why no generalization can be made for EM threshold and iterations. However, in terms of number of states (minimum and maximum) and regularization strength it can be said generally that a minimum of 20 states and maximum of 65 states with a regularization strength of 0.2 and 0.3 would be optimal. Nevertheless, the runtimes with the events less than 2000 were under five hours thus, different parameters of RegPFA can be tried quite easily.

As it can be seen from all the findings that the optimal parameters and runtime was de-

pendent on number of events and their types, probably because of the EM convergence in RegPFA is linearly dependent on the number of events (Breuker et al. 2016). *Please note that these were the findings of this work with UniConnect but can be changed for a different CSCW system but, still they will definitely serve as a good starting point.*

Chapter 5

Conclusion

This last chapter summarizes this research by addressing how the research goals were identified and achieved. Then it points out the advantages and limitations of probabilistic process modelling in CSCW system. Finally, this chapter outline some directions for future work.

5.1 Summary

This work starts with the process mining and identifies how process mining is getting prevalent in the CSCW system van der Aalst (2007). Moreover, it also recognizes that predicting user behaviour in CSCW system is also rising up (Yu et al. 2017). Hence, this research explored the possibilities of probabilistic process modelling in context of CSCW systems and tried to predict user behaviour from the probabilistic model achieved after mining. Since, process mining does not provide any real time information about the process instances that are in action, requirement of probabilistic models were determined. After analyzing several probabilistic process models RegPFA was chosen as the leading one because it outperforms them specially in the context of CSCW system. As CSCW systems often have incomplete logs van der Aalst (2007) and RegPFA handles the problem of *log incompleteness* very efficiently Breuker et al. (2016). Therefore, to

explore the possibilities of probabilistic process mining in CSCW system RegPFA was chosen. After setting the tool and research goals it was important to decide how to fulfill them. The research goals were fulfilled by applying the RegPFA to a CSCW system called UniConnect (Section 2.1). To fully explore the possibilities of probabilistic modelling in CSCW various models with different input configurations and RegPFA parameters were build (Chapter 4) and when possible, every model also tries to predict the user behaviour from the discovered probabilistic process model. Together with exploring and predicting user behaviour every model finds out the optimal parameters for the RegPFA in terms of the number of input configuration (such as number of cases, events). These findings are given in section 4.9 and can be used for the future work if RegPFA has to be applied to some another CSCW system. These findings will save a plenty of time as the runtime of all the models in chapter four was around four to five months. Apart from this, several advantages and limitations were found while applying RegPFA to CSCW system. These advantages and limitations are given below:

- RegPFA can generate process models that can predict the user behaviour quite efficiently but if the discovered process model is too complex i.e, if there are so many edges in the transition system produced then very low pruning ratios has to be applied and that leads to huge information loss as well as underfitted models for example, model one (section 4.3) and model 3 (section 4.5).

- RegPFA can be very efficient in terms of runtime and behaviour prediction, when applied according to the scenario. For example, to have a probabilistic model only for a certain user group in CSCW system (i.e; frequent user or user belong to some community, etc.). In this case the runtime and predictions both will be very efficient and decisions can easily be taken at runtime.

- CSCW system have often have less structured processes and that is why they record every type of event where most of these events are not necessary for the analysis i.e., comes under the noise and if a process is modelled with noise it can easily leads to overfitted model.

- One big advantage of RegPFA was that it handles the issue of incompleteness in logs very well and CSCW logs are often incomplete which can also leads to produce overfitted model but RegPFA handle it really well with the help of Bayesian regularization.

- As CSCW system have huge number of events, the runtime of RegPFA increases as the number of events increase because EM convergence is dependent linearly on the number of events and that is a constraint in the scalability of RegPFA in bigger CSCW systems.

- The discovered process models can not be processed manually as they are too complex but if they have to be processed then very low pruning ratios has to be applied which can highly effect the process of user behaviour prediction but if a CSCW system is highly unpredictable in that case predictive modelling would be reliable even after loosing much information while pruning.

- Lastly, if a CSCW system is too big then a balance should be maintain between the runtime and parameters of RegPFA otherwise decisions can not be taken at real time as RegPFA will run forever.

Hence, it can be said probabilistic mining can be used for CSCW systems by focusing on certain user groups (i.e., frequent user or user belong to some community, etc.) but if applied to the whole system i.e, with huge number of events its highly likely that RegPFA will not produce a good fitted model.

5.2 Future Work

Three possible directions was found for the future work of RegPFA modeling in CSCW systems.

Distributed processing: Currently RegPFA is not implemented in a distributed fashion such as, MapReduce due to which the runtime of the algorithm is sometimes over two

week and if the algorithm is implemented in distributed way then the each combination of number of states and regularization strength for which EM has to run can be processed on different machines as these combinations are independent from each other. For example, if a model has to be processed for minimum and maximum states of 45 and 50 respectively while having a regularization strength of 0.2 with EM running 500 times, then all the five process models can be processed on five different clusters. Therefore, if the runtime on one machine was five days then with the help of distributed processing it can be reduced to one day. This reduction in runtime will also help in processing of probabilistic mining in very big CSCW systems.

Automated Traversal: In the current implementation, if user behaviour have to be predict from the discovered probabilistic model then it has to be processed manually which is not efficient as the number of states rise, number of edges rises exponentially which is cumbersome for manual processing. If the transition system can be traversed automatically it will help in user behaviour prediction more efficiently as the process model does not have to be pruned with very low pruning ration. Moreover, it will also avoid the risk of underfitted model. Furthermore, predictive modelling can then easily be used with bigger CSCW system instead of confining just to certain user group of CSCW system.

Annotated transition system: If the transition system is annotated then the user behaviour can be predicted more accurately as the probability of each transition from one state to another will be known.

Bibliography

Agrawal, R., Gunopulos, D. & Leymann, F. (1998), Mining process models from workflow logs, *in* 'International Conference on Extending Database Technology', Springer, pp. 467–483.

Breuker, D., Matzner, M., Delfmann, P. & Becker, J. (2016), 'Comprehensible predictive models for business processes.', *MIS Quarterly* **40**(4).

Carstensen, P. H. & Schmidt, K. (1999), Computer supported cooperative work: New challenges to systems design, *in* 'In K. Itoh (Ed.), Handbook of Human Factors', Citeseer.

Chen, H., Chiang, R. H. & Storey, V. C. (2012), 'Business intelligence and analytics: From big data to big impact.', *MIS quarterly* **36**(4).

Conforti, R., La Rosa, M. & ter Hofstede, A. H. (2015), 'Noise filtering of process execution logs based on outliers detection'.

Cook, J. E. & Wolf, A. L. (1998), 'Discovering models of software processes from event-based data', *ACM Transactions on Software Engineering and Methodology (TOSEM)* **7**(3), 215–249.

Datta, A. (1998), 'Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches', *Information Systems Research* **9**(3), 275–301.

De La Higuera, C. (2005), 'A bibliographical study of grammatical inference', *Pattern recognition* **38**(9), 1332–1348.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the royal statistical society. Series B (methodological)* pp. 1–38.

Dongen, B., Crooy, R. & Aalst, W. (2008), Cycle time prediction: When will this case finally be finished?, *in* 'Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems:', Springer-Verlag, pp. 319–336.

Günther, C. W. & Rozinat, A. (2012), 'Disco: Discover your processes.', *BPM (Demos)* **940**, 40–44.

Günther, C. W. & Verbeek, E. (2009), 'Xes standard definition', *Fluxicon Process Laboratories* **13**, 14.

Hastie, T., Tibshirani, R. & Friedman, J. (2002), 'The elements of statistical learning: Data mining, inference, and prediction', *Biometrics* .

Herbst, J. (2000*a*), Dealing with concurrency in workflow induction, *in* 'European Concurrent Engineering Conference. SCS Europe'.

Herbst, J. (2000*b*), A machine learning approach to workflow management, *in* 'ECML', Vol. 1810, Springer, pp. 183–194.

Herbst, J. & Karagiannis, D. (2004), 'Workflow mining with inwolve', *Computers in Industry* **53**(3), 245–264.

Jeong, H., Biswas, G., Johnson, J. & Howard, L. (2010), Analysis of productive learning behaviors in a structured inquiry cycle using hidden markov models, *in* 'Educational Data Mining 2010'.

Land, S. & Fischer, S. (2012), 'Rapidminer 5', *Rapid-I Gmbh* .

Ma, H. (2007), 'Process-aware information systems: Bridging people and software through process technology', *Journal of the Association for Information Science and Technology* **58**(3), 455–456.

Maggi, F. M., Di Francescomarino, C., Dumas, M. & Ghidini, C. (2014), Predictive monitoring of business processes, *in* 'International Conference on Advanced Information Systems Engineering', Springer, pp. 457–472.

Mans, R., van der Aalst, W. M. & Verbeek, H. E. (2014), Supporting process mining workflows with rapidprom., *in* 'BPM (Demos)', p. 56.

Mills, A. J., Durepos, G. & Wiebe, E. (2010), *Encyclopedia of case study research: L-z; index*, Vol. 1, Sage.

Provost, F. & Fawcett, T. (2013), *Data Science for Business: What you need to know about data mining and data-analytic thinking*, " O'Reilly Media, Inc.".

Schonenberg, H., Weber, B., Van Dongen, B. & Van der Aalst, W. (2008), Supporting flexible processes through recommendations based on history, *in* 'International Conference on Business Process Management', Springer, pp. 51–66.

Shields, P. & Rangarajan, N. (2013), *A Playbook for Research Methods: Integrating Conceptual Frameworks and Project Management*.

Shmueli, G. & Koppius, O. (2011), 'Predictive analytics in information systems research', *MIS Quarterly* **35**(3), 553–572.

Syri, A. (1997), Tailoring cooperation support through mediators, *in* 'Proceedings of the Fifth European Conference on Computer Supported Cooperative Work', Springer, pp. 157–172.

Van, D. (2011), 'Process mining discovery, conformance and enhancement of business processes'.

Van Der Aalst, W. & Adriansyah (2011), Process mining manifesto, *in* 'International Conference on Business Process Management', Springer, pp. 169–194.

van der Aalst, W. M. (2007), 'Exploring the cscw spectrum using process mining', *Advanced Engineering Informatics* **21**(2), 191–199.

Van Der Aalst, W. M., Reijers, H. A. & Song, M. (2005), 'Discovering social networks from event logs', *Computer Supported Cooperative Work (CSCW)* **14**(6), 549–593.

van der Aalst, W. M., Reijers, H. A., Weijters, A. J., van Dongen, B. F., De Medeiros, A. A., Song, M. & Verbeek, H. (2007), 'Business process mining: An industrial application', *Information Systems* **32**(5), 713–732.

Van der Aalst, W. M., Schonenberg, M. H. & Song, M. (2011), 'Time prediction based on process mining', *Information systems* **36**(2), 450–475.

Van der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E. & Weijters, T. (2009), 'Prom: The process mining toolkit.', *BPM (Demos)* **489**(31), 2.

Van der Aalst, W. M., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G. & Weijters, A. J. (2003), 'Workflow mining: A survey of issues and approaches', *Data & knowledge engineering* **47**(2), 237–267.

Van der Aalst, W., Weijters, T. & Maruster, L. (2004), 'Workflow mining: Discovering process models from event logs', *IEEE Transactions on Knowledge and Data Engineering* **16**(9), 1128–1142.

Van Dongen, B. F., de Medeiros, A. K. A., Verbeek, H., Weijters, A. & Van Der Aalst, W. M. (2005), The prom framework: A new era in process mining tool support., *in* 'ICATPN', Vol. 3536, Springer, pp. 444–454.

Verwer, S., Eyraud, R. & De La Higuera, C. (2014), 'Pautomac: a probabilistic automata and hidden markov models learning competition', *Machine learning* **96**(1-2), 129–154.

Weber, P., Bordbar, B. & Tino, P. (2013), A principled approach to mining from noisy logs using heuristics miner, *in* 'Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on', IEEE, pp. 119–126.

Weijters, A. J. & Van der Aalst, W. M. (2003), 'Rediscovering workflow models from event-based data using little thumb', *Integrated Computer-Aided Engineering* **10**(2), 151–162.

Yu, B., Ren, Y., Terveen, L. G. & Zhu, H. (2017), Predicting member productivity and withdrawal from pre-joining attachments in online production groups., *in* 'CSCW', pp. 1775–1784.