Test On Real life data

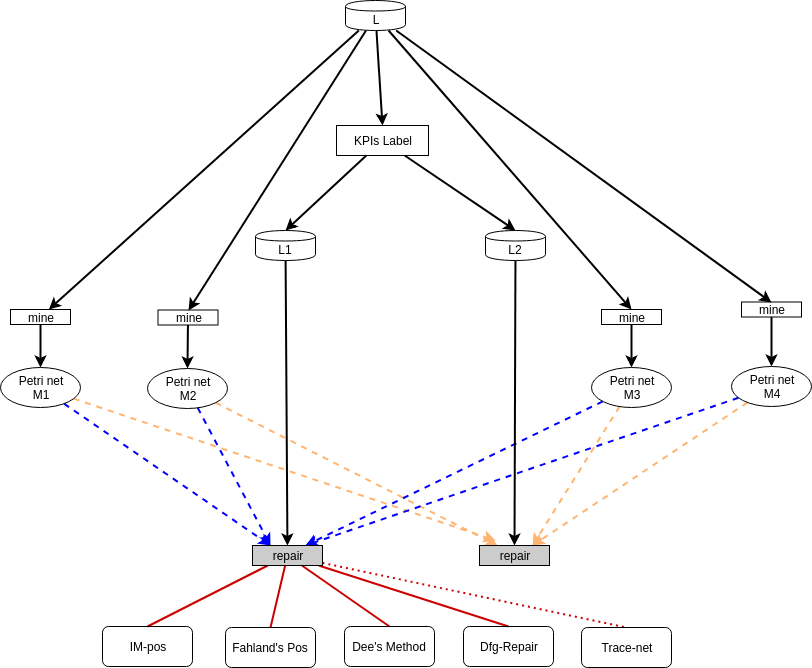
The goal to test on real life data::

– to prove the use of our method with comparison with Dee’s ones, Fahhland’s method

<1> model simplicity and precision

<2> model KPI improvement, how to validate this?? By checking the positive output percentage, it is a measurement.

Test data description::

The first data set is from BPI15\_1.xes, <https://data.4tu.nl/repository/uuid:a0addfda-2044-4541-a450-fdcc9fe16d17>

The information about it :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data ID | File name | process | cases | events | Event class |
| 1 | BPIC15\_1.xes.xml | Real life data from BPI15 | 1199 | 52217 | 398 |
| 2.1 | BPI15\_1\_50\_filter.xes | Heuristic filtering by setting 50% for all from data 1 | 592 | 14307 | 28 |
| 2.2 | BPI15\_1\_50\_filter\_tt.xes | Add trace attribute of throughput time on the traces | 592 | 14307 | 28 |
| 3.1 | BPI15\_1\_50\_filter\_sum\_neg.xes | Filter data on trace attribute of SUMleges, if it is over the 70% sum, then it is assigned with negative. | 166 | 4149 | 28 |
| 3.2 | BPI15\_1\_50\_filter\_sum\_pos.xes | Filter data on trace attribute of SUMleges, if it is below the 70% sum, then it is assigned with positive. | 426 | 10158 | 28 |
| 3.3 | BPI15\_1\_50\_filter\_sum\_labels.xes | Union 3.1 and 3.2 to get an event log with sum labels for our methods | 592 | 14307 | 28 |
| 4.1 | BPI15\_1\_50\_filter\_tt\_neg.xes | Filter data on trace attribute of throughput time, if it is over the 70%, then it is assigned with negative. | 179 | 4360 | 28 |
| 4.2 | BPI15\_1\_50\_filter\_tt\_pos.xes | Filter data on trace attribute of throughput time, if it is over the 70%, then it is assigned with negative. | 413 | 9947 | 28 |
| 4.3 | BPI15\_1\_50\_filter\_tt\_labels.xes | Union 3.1 and 3.2 to get an event log with throughput time labels for our methods | 592 | 14307 | 28 |
| 5 | BPI15\_1\_50\_filter\_18\_classes.xes | Filter 2.1 again with heuristic filter , setting: 80-80-70, | 567 | 10050 | 18 |

Besides of the event logs, we need multiple models as our reference models. We can generate them in the following ways:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Data file | Setting | Model file | Model figures | Description |
| M1 | BPI15\_1\_50\_filter\_18\_classes.xes | Inductive Mine:  IM-infrequent : 0.2  concept: name | BPI\_1\_M1\_18\_classes.pnml | BPI\_1\_M1\_figure.pdf | Have several changes but not so much |
| M2 | BPI15\_1\_50\_filter\_18\_classes.xes | Inductive Mine:  IM-infrequent : 0.5  concept: name | BPI\_1\_M2\_18\_classes.pnml | BPI\_1\_M2\_figure.pdf | It is already like a sequential model  So not go deeper into it |
| M3 | BPI15\_1\_50\_filter.xes | Inductive Mine:  IM-infrequent : 0.2  concept: name | BPI\_1\_M3\_28\_classes.pnml | BPI\_1\_M3\_figure.pdf | Almost in linear but with silent transitions and parallel |
| M3 | BPI15\_1\_50\_filter.xes | Inductive Mine:  IM-infrequent : 0.5  concept: name | BPI\_1\_M4\_28\_classes.pnml | BPI\_1\_M4\_figure.pdf | n linear with few silent transitions and parallel |

After obtaining logs and models, we conduct the test with several techniques.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Log | Model | Techniques | Result |
| T1.1 | 3.2 | M1 | IM\_pos: only on positive, test based on the train data D3.3 | TP: 137 FP: 48 TN:118 FN:289  Recall: 0.3215962441314554  Precision: 0.7405405405405405  Accuracy: 0.43074324324324326  F-score: 0.44844517184942717 |
|  | 3.2 | M1 | Fahland’s method: only on positive, test based on train data |  |
|  | 3.3 | M1 | Dees method: on whole data |  |
|  | 3.3 | M1 | Dfg method: on whole data |  |
|  | 3.2 | M1 | Trace nets |  |
| T1.2 | 3.2 | M2 |  |  |
|  | ... | ... |  |  |
| T1.3 | 3.2 | M3 |  |  |
|  | ... | ... |  |  |
| T1.4 | 3.2 | M4 |  |  |
|  | ... | ... |  |  |
| T2.1 | 4.2 | M1 |  |  |
|  | ... | ... |  |  |
| T2.1 | 4.2 | M2 |  |  |
|  | ... | ... |  |  |
| T2.3 | 4.2 | M3 |  |  |
|  | ... | ... |  |  |
| T2.4 | 4.2 | M4 |  |  |
|  | ... | ... |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

The total number of tests for one event logs BPI\_1.xes is 2\*4\*5=40; we can repeat the method on the other logs BPI\_2.xes, BPI\_3.xes,BPI\_4.xes,BPI\_5.xes. So total number is 40\*5 =200… If we can conduct this tests.

But first make sure that this methods work