Improving Process Mining prediction results in processes that change over time

Alessandro Berti
Email: alessandro.berti89@gmail.com

***Abstract*—In this paper I propose a method in order to improve the accuracy of predictions, related to incomplete traces, in event logs that record changes in the underlying process. These "second-order dynamics" hamper the functioning of Process Mining discovery algorithms, but also hamper prediction results. The method is simple to implement as it is based exclusively on the Control Flow perspective and is computationally efficient. The approach has been validated on the BPI Challenge 2015's Municipality 5 event log, that contains an interesting shift in the process due to the union of the municipality with another municipality.**

I. Introduction

Business processes are constantly evolving to adapt to new opportunities, and continuous improvement is needed for a company in order to remain at the top. In some cases, the quality of a process can be measured in time: for example, in Service Desk tickets, avoiding to break S.L.A. is important. Knowing in advance which instances are most critical, assigning more resources to them, may be vital for some organizations.

Here arise the need of a good prediction algorithm for process instances. Process Mining provides some techniques to predict the completion time of instances, however they assume the underlying process to be static, obtaining in many cases poor results. In this paper I provide an approach to improve existing prediction results by considering the fact that the process change over time. Assessment done on BPI Challenge 2015's Municipality 5 event log shows that the approach actually improved the prediction results in a process that changed over time.

II. Background

Process Mining [1] is a relatively new discipline that aims to automatically discover and measure things about processes. It mainly uses automatic recordings of events, that are called event logs. Sub-disciplines of Process Mining are process discovery [2], that aims to automatically discover the process schema starting from an event log, process conformance [3] that is useful to see differences between a de-jure process model and the current executions of the process (recorded in the event log), process performance [2] that wants to identify bottlenecks inside business processes starting from event logs, and process-related predictions which I'll analyze later. Event logs are organised in traces, that are collections of events serving to a particular purpose. For example, a trace might regard a single case served by an Help Desk process. Meanwhile, events can be described by several attributes, including:

- The activity that has been performed.
- The originator of the event (the organizational resource that has done the event).
- The timestamp (the time in which the event has been executed).
- The transition of the event, that refers to the state of execution (a "complete" transition means that the activity actually ended, a "start" transition means that the activity started).

The trace itself can be characterised by several attributes (for example, in an Help Desk system, the severity of the case might be an attribute). I can consider as start timestamp of the trace the minimum timestamp of its events, and as end timestamp the maximum timestamp of its events. Many times, there is only a transition ("complete"), so the trace might be described (in the Control Flow perspective) by the succession/list of its activities. This is a condition required by some Process Discovery algorithms, like the Heuristics Miner [4], that aims to discover the process schema by calculating the dependency between activities. This means that if in all occurrences of an event log an activity (1) is followed by another activity (2), then Heuristics Miner can discover a process schema in which activity 1 is always followed by activity 2. So, Heuristics Miner analyze[1] the paths in a trace: a path is a direct succession of activities in a trace. For example, if a trace contains (analyzing only the Control Flow perspective) the following activities: ABCDE; then all the paths contained in the trace are: AB BC CD DE. I say that a path belongs to a trace if it is contained in the trace. An important definition I provide for later use is about the belonging of a trace to a time interval. I can say that a trace, with *start* as the start timestamp and *end* as the end timestamp, belongs to a time interval $[t_1, t_2]$, if one of the following three conditions is satisfied:

1) $start \leq t_1 \leq t_2 \leq end$
2) $t_1 \leq start < t_2$
3) $t_1 < end \leq t_2$

It might be important also to consider the difference between complete and incomplete traces. The last ones are being reported in the log, although their execution is not finished. The distinction is somewhat difficult to make, I can refer to [5] for further discussion. A possible way to detect incomplete traces is about applying heuristics to the end activities: if the end activity of a trace can be found as an end activity in many other traces, then I can consider it to be a complete trace, otherwise incomplete. The succession of the activities

[1] Among the others

DiffInt(*log*,I_1,I_2)
Require: An event log *log*, time sub-intervals I_1 and I_2.
$Tr_1 = \{tr \in log, tr \in I_1\}$
$Tr_2 = \{tr \in log, tr \in I_2\}$
$RelImp_1 = \left\{\left(path, \frac{\#occ.\ path.}{\#Tr_1}\right) | \exists tr \in Tr_1, path \in tr\right\}$
$RelImp_2 = \left\{\left(path, \frac{\#occ.\ path.}{\#Tr_2}\right) | \exists tr \in Tr_2, path \in tr\right\}$
$AllPaths = \pi_0(RelImp_1) \cup \pi_0(RelImp_2)$
$D = \{\}$
for $P \in AllPaths$ **do**
  if $P \in \pi_0(RelImp_1)$ *and* $P \in \pi_0(RelImp_2)$ **then**
    $D[P] = \frac{Abs(RelImp_1[P] - RelImp_2[P])}{Max(RelImp_1[P], RelImp_2[P])}$
  else
    $D[P] = 1$
  end if
end for
return D

Fig. 1. The algorithm to calculate the difference between the paths' importance in two different sub-intervals. It starts calculating the relative importance of paths in the two sub-intervals and then the difference between the sub-intervals.

Importance(tr,D)
Require: A complete trace tr, difference of importance of paths between intervals D.
return $avg_{(A_1,A_2) \in Paths(tr)}\{1 - D[(A_1, A_2)]\}$

Fig. 2. The algorithm to calculate the importance of a trace tr in the difference of temporal contexts described by the dictionary D. I get a higher importance if the temporal context of the two traces is similar, while I get a lower importance if the process underlying the two sub-intervals is noticeably different.

Similarity(*log*,tr_1,tr_2,*intervals*)
Require: An event log *log*, an incomplete trace tr_1, a complete trace tr_2 (used for prediction purposes), collection of time sub-intervals *intervals*.
if $\exists I \in intervals | tr_1 \in I, tr_2 \in I$ **then**
  return 1
end if
return $\max_{I_1, I_2 \in intervals | tr_1 \in I_1, tr_2 \in I_2} Importance(tr = tr_2, D = DiffInt(log, I_1, I_2))$

Fig. 3. The algorithm to calculate the similarity between the temporal context of two traces, one of them is incomplete and the other is complete and used for prediction. I am searching the couple of time sub-intervals that satisfy the following conditions: the first trace belongs to the first sub-interval, the second trace belongs to the second sub-interval; and minimize the average difference of importance of paths contained in the second trace (belonging to the second sub-interval) given the first sub-interval. This step is useful to calculate the importance of the complete trace in the temporal context of the incomplete trace, giving a weight to it. This gives less importance to traces that describe a different underlying process.

of an incomplete trace might be shared with a complete trace, being a "prefix". An interesting task about incomplete traces might be the prediction of their attributes. For previous work about prediction tasks, I can refer to [6], that mainly describes a method for the prediction of the remaining time of incomplete traces. Basically, the idea is to build an annotated "transition system"[2], that is learned from previous executions, i.e. complete traces, using an abstraction mechanism.

A further step is the one explained in [7]. The prediction of the remaining time is calculated using these two factors:

- The likelihood of following activities, given the data collected so far.
- The remaining time estimation given by a regression model built upon the data.

Basically, this method is an improvement over [6] because it considers not only the Control Flow perspective, but also other events' attributes, identifying the ones that are relevant to the prediction of the remaining time. More attributes you have, more accurate the prediction should be. A process specialist could insert artificial attributes to events (for example, the workload of the resources, or the work in process), in order to further improve the prediction. However, an aspect somewhat ignored in predictions is the fact that the underlying process might change during time. As [8] reports, changes might induce one of the following drifts:

- Recurring drifts: these ones refer to changes that happen in some moments of the year (seasonal influence) or some other recurring changes (for example, a financial process might change near the financial closure of the year).
- Sudden drifts: these refer to big changes in the process: the "old" process cease to exist, while a "new" process starts to be applied.
- Gradual drifts: these refer to a gradual shift from an "old" process to a "new" process. This might be

[2] I skip the explanation of this concept, as it's not firmly connected with the explanation of my method. For further discussion, see [6].

done to let the organizational resources learn the new process.

A method to identify and to cope with changes in the process is described always in [8]: at a first time you have to identify change points in the process (i.e. the times when the process is different), after that you have to identify the region of the change and the type of the change (recurring, sudden, gradual drifts). The last step is about exploiting this knowledge to "unravel" the evolution of the process, describing the change process. Basically, an application of the classical Process Discovery algorithms (for example Heuristics Miner [4], Inductive Miner [9]) can be reliable only in time intervals that contains a consistent, without-drifts process. The same is valid for the prediction algorithms, as a prediction based indifferently on the entire process (that might be changed meanwhile) is not-so-accurate. However, also a prediction based only on the last iteration of the process might be incomplete and not-so-accurate.

III. Method

My method wants to overcome the limitations of both a prediction based indifferently on the entire process, and a prediction based only on the last iteration of the process (it might be a restricted time interval). I do not propose a method to detect change points and analyze them, for this scope I refer to [8], [10]; I start from the assumption to know where change points are[3]. Starting from the overall time interval of events contained in an event log, I suppose there is a collection of time sub-intervals covering the entire time interval and in which the underlying process is constant.

The method is based on the knowledge of a distance measure between two time sub-intervals. This way, you have a method to say how much reliable a complete trace (that might be following a slightly different process) is in the prediction of an incomplete trace that is based on the last iteration of the process. The proposed Algorithm 1 measures the distance path by path, as some paths might be equally present in both intervals. Algorithm 1 basically works calculating the relative importance of each path in each of the subintervals (that is the ratio of the number of path's occurrences and the number of traces), and then comparing this quantity between the intervals. The reliability of the trace in the context of a prediction can be then calculated using Algorithm 2. I propose to use the average (done on all the paths of a trace) of the distance calculated using Algorithm 1. Other statistics (like the maximum of the distance) proved to be less reliable.

Algorithm 3 uses the previous two algorithms, starting from a couple of traces (the first of them is the one I want to predict), the event log and the subdivision in sub-intervals. It tries to find two sub-intervals, containing respectively the two traces[4], that are at a minimum distance, so maximising the similarity. This has been done in order to avoid giving unnecessary low weights of similarity to traces whose duration has been longer than the sub-intervals in which the underlying process is constant.

[3]This could also be done with an interview to organizational resources

[4]With the meaning explained in Background

Then, to obtain the prediction, one could use van der Aalst's [6] algorithm, weighting the traces used for the prediction through Algorithm 3.

IV. Results

The proposed algorithms have been tested on the BPI Challenge 2015's Municipality 5 event log[5]. The log describes a very complex process, with many activities, and is particularly interesting because this municipality (Municipality 5) got merged with another municipality (Municipality 2[6]) at a certain point of time, and the process became different. I can roughly identify some different time intervals:

1) The first one, going from the start of the log to June 2013, in which Municipality 5 was substantially departed from Municipality 2.
2) The shift one, going from June 2013 to June 2014, in which Municipality 5 get merged with Municipality 2.
3) The second one, going from June 2014 to the end of the log, in which Municipality 5 is already united with Municipality 2.

These sub-intervals were identified with a resource analysis, seeing that the resources working in the process got more numerous, and the point of the shift is comprised between June 2013 and June 2014. Being these sub-intervals roughly identified, the shift interval will be ignored for prediction purposes, and I'll focus on the first and the second interval, in which the underlying process is different.

I used van der Aalst's [6] as prediction[7] algorithm, weighting the traces used for the prediction using Algorithm 3. I considered all the traces in the log as completed ones, so for the prediction purposes I took a prefix of each one, predicted the completion time and compared it to the effective completion time. The effectiveness of the prediction was measured using two standard measures (MAPE and RMSPE), briefly explained below. Here A_i is relative to the actual value (the effective completion time) and F_i is relative to the predicted completion time.

$$\mathit{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right|$$

$$\mathit{RMSPE} = \frac{\sqrt{\sum_{i=1}^{n}(A_i - F_i)^2}}{n}$$

In Table I there are some results of the application of Algorithm 1 to Municipality 5 event log. You can see that for some paths there is a big difference in importance between intervals. This reflects the big change in the underlying process.

In Table II you can see some results related to predictions. I have compared three different conditions:

- The prediction (of the remaining time) relative to a prefix of a trace (belonging indifferently to the first or

[5]DOI 10.4121/uuid:b32c6fe5-f212-4286-9774-58dd53511cf8

[6]DOI 10.4121/uuid:63a8435a-077d-4ece-97cd-2c76d394d99c

[7]Prediction of the remaining time

Succ. of act.	Count(1)	Rel.Imp.(1)	Count(2)	Rel.Imp.(2)	Diff.Imp.
01_HOOFD_011,01_HOOFD_012	362	0.4707	155	0.6078	0.2256
01_HOOFD_490_1,01_HOOFD_490_2	295	0.3836	1	0.0039	0.9898
01_HOOFD_480,01_HOOFD_490_1	275	0.3576	1	0.0039	0.9890
01_HOOFD_030_1,08_AWB45_020_2	194	0.2523	1	0.0039	0.9845
01_HOOFD_370,01_HOOFD_375	185	0.2406	1	0.0039	0.9837
01_HOOFD_030_2,01_HOOFD_015	182	0.2367	3	0.0118	0.9503
01_HOOFD_330,09_AH_I_010	178	0.2315	119	0.4667	0.5040
01_HOOFD_380,01_HOOFD_430	170	0.2211	1	0.0039	0.9823
01_HOOFD_195,01_HOOFD_250_1	163	0.2120	8	0.0314	0.8520
01_HOOFD_050,04_BPT_005	139	0.1808	51	0.2000	0.0962
08_AWB45_005,08_AWB45_010	137	0.1782	5	0.0196	0.8899
04_BPT_005,01_HOOFD_065_1	137	0.1782	1	0.0039	0.9780
01_HOOFD_250_2,01_HOOFD_330	134	0.1743	1	0.0039	0.9775
01_HOOFD_101,01_HOOFD_180	124	0.1612	1	0.0039	0.9757
02_DRZ_010,01_HOOFD_050	122	0.1586	1	0.0039	0.9753
01_HOOFD_196,01_HOOFD_200	117	0.1521	12	0.0471	0.6907
04_BPT_005,01_HOOFD_061	116	0.1508	1	0.0039	0.9740
01_HOOFD_065_0,01_HOOFD_061	107	0.1391	2	0.0078	0.9436
13_CRD_010,01_HOOFD_480	98	0.1274	141	0.5529	0.7695
08_AWB45_005,01_HOOFD_196	95	0.1235	28	0.1098	0.1112
01_BB_540,01_BB_775	92	0.1196	14	0.0549	0.5411
01_HOOFD_510_0,01_BB_540	92	0.1196	1	0.0039	0.9672
01_HOOFD_010,01_HOOFD_030_2	88	0.1144	2	0.0078	0.9315
08_AWB45_010,08_AWB45_020_0	88	0.1144	59	0.2314	0.5054
01_HOOFD_490_4,01_HOOFD_500	82	0.1066	2	0.0078	0.9264

TABLE I. DIFFERENCE IN IMPORTANCE OF SEVERAL PATHS IN THE DIFFERENT INTERVALS. THIS REFLECTS THE CHANGE IN THE UNDERLYING PROCESS. THE FIRST COLUMN DESCRIBES THE PATH, THE SECOND AND THE FOURTH REPORT THE COUNT OF THE PATHS IN THE RESPECTIVE TIME INTERVALS, THE THIRD AND THE FIFTH REPORT THE RELATIVE IMPORTANCE (THE AVERAGE OF THE OCCURRENCES OF PATHS INSIDE TRACES). THE SIXTH COLUMN IS THEN CALCULATED AS THE RATIO OF THE ABSOLUTE DIFFERENCE OF THE RELATIVE IMPORTANCES AND THE MAXIMUM OF THE TWO RELATIVE IMPORTANCES.

Start of trace	N. of trac.(1+2)	MAPE(1+2)	RMSPE(1+2)	MAPE(1)	RMSPE(1)	MAPE(2)	RMSPE(2)
01_HOOFD_010,01_HOOFD_011	512	92.8544	7596039.2797	**72.1547**	**5758334.0360**	**29.4204**	**3684134.2904**
01_HOOFD_010,01_HOOFD_030_2	178	3.5637	9587638.9410	3.5637	9587638.9410	**1.0269**	**497670.2839**
01_HOOFD_010,01_HOOFD_015	89	0.7490	7087620.4087	0.7490	7087620.4087	0.9472	**537482.9933**
01_HOOFD_010,01_HOOFD_065_2	51	0.3856	4639109.3841	0.3856	4639109.3841	0.9343	**733715.3557**
01_HOOFD_010,01_HOOFD_020	45	6466.2098	5150725.7065	**5897.2386**	**4897063.7661**	**1157.3147**	**1237980.3061**
01_HOOFD_010,02_DRZ_010	13	21.1334	6483207.2943	**5.0226**	**2740205.8570**	**20.1936**	**5823588.8817**
01_HOOFD_030_2,01_HOOFD_010	11	1.4041	30442608.9241	**1.3705**	**28448527.4061**	**0.8268**	**6779677.5687**
01_HOOFD_011,01_HOOFD_020	8	0.8641	3540610.0560	**0.5266**	**2475832.0442**	**0.6885**	**2767641.7088**
01_HOOFD_010,01_HOOFD_100	7	116.1590	49573665.1192	**53.6666**	**45699111.1204**	**4.6097**	**9047583.6599**
01_HOOFD_010,08_AWB45_020_2	6	0.4237	2882146.0787	0.4237	2882146.0787	0.8300	**2054337.2500**
01_HOOFD_065_2,01_HOOFD_010	4	1.0459	8758215.5171	1.0459	8758215.5171	**0.9356**	**3509933.2780**
01_HOOFD_010,04_BPT_005	3	48.0765	6306318.0537	**7.8786**	**2877381.7631**	48.0765	6306318.0537
01_HOOFD_010,01_HOOFD_180	2	0.3875	3894038.0000	0.7555	5147446.2190	0.3875	3894038.0000
01_HOOFD_010,01_HOOFD_190_2	2	3.7315	114725504.0000	3.7315	114725504.0000	**0.9044**	**47393188.8894**
01_HOOFD_460,01_HOOFD_010	2	0.0496	1410228.0000	0.0496	1410228.0000	0.9016	14055460.3380
01_HOOFD_065_2,01_HOOFD_100	2	0.8069	1443136.0000	0.8069	1443136.0000	0.9809	**1114929.5283**

TABLE II. RESULTS RELATED TO THE PREDICTION OF REMAINING TIME OF TRACES WHEN I KNOW THE ACTIVITIES AND THE TIMESTAMPS OF THE FIRST TWO EVENTS OF A TRACE (IN THE BPI CHALLENGE 2015'S MUNICIPALITY 5 LOG; THE CONSIDERED TIME INTERVALS ARE SPECIFIED IN THE MAIN TEXT). IN THE THIRD COLUMN, I HAVE REPORTED MAPE AND RMSPE OF THE PREDICTION (DONE USING VAN DER AALST'S METHOD, SEE REFERENCE [6]) THAT TAKES IN ACCOUNT BOTH THE TIME INTERVALS. IN THE FOURTH AND THE FIFTH COLUMN, I HAVE REPORTED MAPE AND RMSPE OF THE PREDICTIONS (DONE USING THE VAN DER AALST'S METHOD) IN WHICH THE WEIGHTING OF THE TRACES TAKES IN ACCOUNT (USING MY METHOD DESCRIBED IN ALGORITHM 3) THE CHANGE IN THE UNDERLYING PROCESS. MORE SPECIFICALLY, IN THE FOURTH COLUMN I HAVE REPORTED THE RESULTS OF THE PREDICTIONS OF THE REMAINING TIME OF TRACES BELONGING TO THE FIRST TIME INTERVAL, WHILE IN THE FIFTH COLUMN I HAVE REPORTED THE RESULTS OF THE PREDICTIONS RELATED TO TRACES BELONGING TO THE SECOND TIME INTERVAL. I CAN SEE THAT IN A GOOD NUMBER OF CASES PREDICTIONS IMPROVED.

Start of trace	N. of trac.(1+2)	MAPE(1+2)	RMSPE(1+2)	MAPE(1)	RMSPE(1)	MAPE(2)	RMSPE(2)
01_HOOFD_010,01_HOOFD_011,01_HOOFD_020	482	97.8206	7609140.4442	**74.4032**	**5684585.8721**	**32.2999**	**3779530.8508**
01_HOOFD_010,01_HOOFD_030_2,01_HOOFD_015	122	14930.5165	10071033.0171	14930.5165	10071033.0171	**424.3935**	**411018.8551**
01_HOOFD_010,01_HOOFD_015,01_HOOFD_020	88	0.7528	7113510.0065	0.7528	7113510.0065	0.9469	**544560.6131**
01_HOOFD_010,01_HOOFD_030_2,01_HOOFD_065_2	35	0.5755	6048120.8883	0.5755	6048120.8883	0.9416	**742048.0423**
01_HOOFD_010,01_HOOFD_020,03_GBH_005	32	0.5460	5410643.3730	0.5460	5410643.3730	0.8779	**907541.9461**
01_HOOFD_010,01_HOOFD_011,01_HOOFD_015	25	2.1717	8279672.0234	2.1717	8279672.0234	**0.8312**	**1897717.0886**
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_030_2	24	0.3386	3596116.9073	0.3386	3596116.9073	0.9599	**758544.5811**
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_011	10	0.2136	1730206.6639	0.2136	1730206.6639	0.8376	**1539281.7300**
01_HOOFD_030_2,01_HOOFD_010,01_HOOFD_015	9	1.5834	34502687.4489	1.5834	34502687.4489	**0.8614**	**5178829.1068**
01_HOOFD_010,02_DRZ_010,04_BPT_005	9	36.2173	7428347.0140	**6.8727**	**2504995.6683**	36.2173	7428347.0140
01_HOOFD_010,01_HOOFD_020,01_HOOFD_015	9	28662.2724	3743629.8187	28662.2724	3743629.8187	**1827.3567**	**1201595.5150**
01_HOOFD_010,01_HOOFD_030_2,01_HOOFD_100	8	0.6689	7220627.9116	0.6689	7220627.9116	0.9488	**2202998.4987**
01_HOOFD_010,01_HOOFD_030_2,01_HOOFD_020	6	0.5900	6852557.9972	0.5900	6852557.9972	0.8914	**3524148.5139**
01_HOOFD_010,01_HOOFD_011,01_HOOFD_012	5	4.4923	6541984.7061	**4.3511**	**4989471.1460**	**1.0229**	**5111010.0414**
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_020	5	0.1980	1989049.6259	0.1980	1989049.6259	0.9366	2165896.7836
01_HOOFD_010,08_AWB45_020_2,01_HOOFD_011	5	1.9712	4342968.0844	1.9712	4342968.0844	**0.6074**	**1762686.1021**
01_HOOFD_010,01_HOOFD_030_2,08_AWB45_020_2	5	1.4118	18454190.2346	1.4118	18454190.2346	**0.8784**	**7320269.2102**
01_HOOFD_011,01_HOOFD_020,02_DRZ_010	4	0.9579	5093072.7121	**0.7281**	**2374478.6397**	0.9579	5093072.7121
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_015	4	0.1935	1518991.7059	0.1935	1518991.7059	0.9385	1783136.1910
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_100	3	2.0393	25282013.7310	2.0393	25282013.7310	**0.9485**	**8849956.0681**
01_HOOFD_010,02_DRZ_010,01_HOOFD_011	3	1.9991	3157732.3759	**1.9245**	**2524545.0040**	**0.3542**	**2102105.9026**
01_HOOFD_011,01_HOOFD_020,03_GBH_005	3	1.1284	4899912.1248	1.1284	4899912.1248	**0.6089**	**3021026.5790**
01_HOOFD_010,04_BPT_005,01_HOOFD_065_0	2	0.8638	7474836.0000	**0.7944**	**5532054.4569**	0.8638	7474836.0000
01_HOOFD_065_2,01_HOOFD_010,01_HOOFD_030_2	2	2.6098	25116980.0000	2.6098	25116980.0000	**0.9108**	**11058921.0330**
01_HOOFD_010,01_HOOFD_100,01_HOOFD_065_2	2	0.3275	13389370.0000	0.3275	13389370.0000	0.9616	21201323.5022
01_HOOFD_010,01_HOOFD_100,08_AWB45_020_2	2	1.9679	44208424.0000	1.9679	44208424.0000	**0.9351**	**21257607.3203**
01_HOOFD_010,01_HOOFD_065_2,01_HOOFD_190_2	2	0.0147	144842.0000	0.0147	144842.0000	0.9882	4932713.5820
01_HOOFD_010,01_HOOFD_180,08_AWB45_005	2	0.3875	3894038.0000	0.7555	5147446.2190	0.3875	3894038.0000
01_HOOFD_010,01_HOOFD_020,01_HOOFD_011	2	1.2573	12031792.0000	1.3985	**11188363.6467**	**0.6438**	**8663993.7255**

TABLE III. RESULTS RELATED TO THE PREDICTION OF REMAINING TIME OF TRACES WHEN I KNOW THE ACTIVITIES AND THE TIMESTAMPS OF THE FIRST THREE EVENTS OF A TRACE (IN THE BPI CHALLENGE 2015'S MUNICIPALITY 5 LOG; THE CONSIDERED TIME INTERVALS ARE SPECIFIED IN THE MAIN TEXT). IN THE THIRD COLUMN, I HAVE REPORTED MAPE AND RMSPE OF THE PREDICTION (DONE USING VAN DER AALST'S METHOD, SEE REFERENCE [6]) THAT TAKES IN ACCOUNT BOTH THE TIME INTERVALS. IN THE FOURTH AND THE FIFTH COLUMN, I HAVE REPORTED MAPE AND RMSPE OF THE PREDICTIONS (DONE USING THE VAN DER AALST'S METHOD) IN WHICH THE WEIGHTING OF THE TRACES TAKES IN ACCOUNT (USING MY METHOD DESCRIBED IN ALGORITHM 3) THE CHANGE IN THE UNDERLYING PROCESS. MORE SPECIFICALLY, IN THE FOURTH COLUMN I HAVE REPORTED THE RESULTS OF THE PREDICTIONS OF THE REMAINING TIME OF TRACES BELONGING TO THE FIRST TIME INTERVAL, WHILE IN THE FIFTH COLUMN I HAVE REPORTED THE RESULTS OF THE PREDICTIONS RELATED TO TRACES BELONGING TO THE SECOND TIME INTERVAL. I CAN SEE THAT IN A GOOD NUMBER OF CASES PREDICTIONS IMPROVED.

second time interval), using for the prediction all the traces in the log.

- The prediction relative to a prefix of a trace belonging to the first interval, using for the prediction all the traces weighted accordingly to Algorithm 3.
- The prediction relative to a prefix of a trace belonging to the second interval, using for the prediction all the traces weighted accordingly to Algorithm 3.

I have taken as prefix the first two activities. The results are then grouped based on their prefix.

In Table III the same techniques are applied to a prefix containing the first three activities of the trace. In many occurrences prediction results obtained by weighting the traces using Algorithm 3 are improved in comparison to the classical technique.

V. CONCLUSIONS

In this paper I have proposed a method to take care, in the prediction of traces' attributes, of drifts in the examined process. This method assumes that I already know the times in which the process changes. All these changes, might they be seasonal, gradual or sudden, split the overall time interval into subintervals in which I could assume that the process is constant. The discovery of these times could be done in an automated way, for example using the algorithm described in [8], or manually through an interview. For each time sub-interval you can observe how many times two activities are in direct succession; after that, you could compare the distributions measured in the different sub-intervals. This is useful to understand how much the process is different between different sub-intervals, and to give a different weight to the different (complete) traces one could use to predict the outcome of a incomplete trace. This is useful in each type of prediction, as the prediction of the remaining time in a trace.

The described algorithms are pretty easy to implement, and are not computationally expensive (my implementation has been realised in a plain Python script). However, the approach considers only the control flow perspective, and ignores other perspectives (like the data perspective and the resource perspective) in which the process could change over time. Indeed, changing roles inside an organizational process might change the throughput times, because of different skills, changed workloads and difficulties in collaboration between different work groups. I can cite some literature related to social and work psychology [11], [12], [13], that give insights on how much inter-group relationships are important for organizational performance. Generally, one could identify inter-group distances in a process by measuring times elapsed between activities performed by different roles. This can be related to the Lean Manufacturing concept of Flow Rate [14], [15], [16]. Another aspect is related to the group's Transactive Memory [17], [18], [19]. Transactive Memory is a psychological concept, that I could explain as "group memory" and is related to the specialization and the coordination of the group [20], [21]. Indeed, a change in the work group's structure, that could be motivated by a change in the process, can hamper a lot the group's performance, because of the newcomers' need to know the rest and the roles of the group, or some people exiting the group. It's a pity that Transactive Memory in groups is generally difficult to measure [22], because it's a powerful tool to measure group performance.

There's also scope to research related to non-instantaneous events, that could include several transitions (start, complete, stop, resume) [23], as the framework described here works only for instantaneous events (each trace could be described by a succession of conclusive activities). Overall, my method seems to be good performing on the BPI Challenge's Municipality 5 log. In that log, the process changes after the union with another municipality (Municipality 2). Not in every event log, however, I register a change in the underlying process. In that case, my method is useless.

Moreover, current results related to prediction of attributes (e.g. remaining time) are not that good, even with my improvement. There's something more to come in order to get good and reliable predictions.

References

[1] W. Van Der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs *et al.*, "Process mining manifesto," in *Business process management workshops*. Springer, 2012, pp. 169–194.

[2] W. Van Der Aalst, *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[3] W. M. Van der Aalst and A. K. A. de Medeiros, "Process mining and security: Detecting anomalous process executions and checking process conformance," *Electronic Notes in Theoretical Computer Science*, vol. 121, pp. 3–21, 2005.

[4] A. Weijters, W. M. van Der Aalst, and A. A. De Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, pp. 1–34, 2006.

[5] W. M. Van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. Weijters, "Workflow mining: a survey of issues and approaches," *Data & knowledge engineering*, vol. 47, no. 2, pp. 237–267, 2003.

[6] W. M. Van der Aalst, M. H. Schonenberg, and M. Song, "Time prediction based on process mining," *Information Systems*, vol. 36, no. 2, pp. 450–475, 2011.

[7] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni, "Data-aware remaining time prediction of business process instances," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 816–823.

[8] R. J. C. Bose, W. M. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Handling concept drift in process mining," in *Advanced Information Systems Engineering*. Springer, 2011, pp. 391–405.

[9] S. J. Leemans, D. Fahland, and W. M. van der Aalst, "Discovering block-structured process models from event logs containing infrequent behaviour," in *Business Process Management Workshops*. Springer, 2014, pp. 66–78.

[10] J. Martjushev, R. J. C. Bose, and W. M. van der Aalst, "Change point detection and dealing with gradual and multi-order dynamics in process mining," in *Perspectives in Business Informatics Research*. Springer, 2015, pp. 161–178.

[11] H. Tajfel, "Social psychology of intergroup relations," *Annual review of psychology*, vol. 33, no. 1, pp. 1–39, 1982.

[12] ——, *Social identity and intergroup relations*. Cambridge University Press, 2010.

[13] B. E. Ashforth and F. Mael, "Social identity theory and the organization," *Academy of management review*, vol. 14, no. 1, pp. 20–39, 1989.

[14] R. Shah and P. T. Ward, "Lean manufacturing: context, practice bundles, and performance," *Journal of operations management*, vol. 21, no. 2, pp. 129–149, 2003.

[15] T. Melton, "The benefits of lean manufacturing: what lean thinking has to offer the process industries," *Chemical Engineering Research and Design*, vol. 83, no. 6, pp. 662–673, 2005.

[16] C. Cassell, J. Worley, and T. Doolen, "The role of communication and management support in a lean manufacturing implementation," *Management Decision*, vol. 44, no. 2, pp. 228–245, 2006.

[17] D. M. Wegner, "Transactive memory: A contemporary analysis of the group mind," in *Theories of group behavior*. Springer, 1987, pp. 185–208.

[18] R. L. Moreland and L. Myaskovsky, "Exploring the performance benefits of group training: Transactive memory or improved communication?" *Organizational behavior and human decision processes*, vol. 82, no. 1, pp. 117–133, 2000.

[19] J. R. Austin, "Transactive memory in organizational groups: the effects of content, consensus, specialization, and accuracy on group performance." *Journal of Applied Psychology*, vol. 88, no. 5, p. 866, 2003.

[20] L. Argote, "An opportunity for mutual learning between organizational learning and global strategy researchers: transactive memory systems," *Global Strategy Journal*, vol. 5, no. 2, pp. 198–203, 2015.

[21] C. Heavey and Z. Simsek, "Transactive memory systems and firm performance: An upper echelons perspective," *Organization Science*, 2015.

[22] K. Lewis, "Measuring transactive memory systems in the field: scale development and validation." *Journal of Applied Psychology*, vol. 88, no. 4, p. 587, 2003.

[23] A. Burattin, "Heuristics miner for time interval," in *Process Mining Techniques in Business Environments*. Springer, 2015, pp. 85–95.