On the Feasibility of Distributed Process Mining in Healthcare

Roberto Gatta[1][0000−0002−4716−9925], Mauro Vallati[3][0000−0002−8429−3570], Jacopo Lenkowicz[1], Carlotta Masciocchi[1], Francesco Cellini[2], Luca Boldrini[1], Carlos Fernandez Llatas[4], Vincenzo Valentini[1,2], and Andrea Damiani[1]

[1] Dipartimento di Diagnostica per Immagini, Radioterapia Oncologica ed Ematologia, Istituto di Radiologia Universiá Cattolica del Sacro Cuore, Roma, Italia
roberto.gatta.bs@gmail.com
[2] UOC Radioterapia Oncologica, Fondazione Policlinico Universitario A. Gemelli IRCCS, Roma, Italia
[3] University of Huddersfield, Huddersfield, United Kingdom
m.vallati@hud.ac.uk
[4] ITACA-Universitat Politecnica de Valencia, Valencia, Spain

Abstract. Process mining is gaining significant importance in the healthcare domain, where the quality of services depends on the suitable and efficient execution of processes. A pivotal challenge for the application of process mining in the healthcare domain comes from the growing importance of multi-centric studies, where privacy-preserving techniques are strongly needed.
In this paper, building on top of the well-known Alpha algorithm, we introduce a distributed process mining approach, that allows to overcome problems related to privacy and data being spread around. The introduced technique allows to perform process mining without sharing any patients-related information, thus ensuring privacy and maximizing the possibility of cooperation among hospitals.

Keywords: Process mining · Healthcare · Distributed learning.

1 Introduction

Process Mining (PM) is an emerging topic aimed at building organizational process representations from real data [3]; it allows, for instance, to identify bottlenecks and undesired or unsuspected processes' paths. Healthcare represents a challenging domain for PM: there is a remarkable cultural gap between clinicians and computer scientists stakeholders; each event can have different meanings depending on the clinical perspective, the specific aim, the period of time or the patient it refers to. Furthermore, the psychology of patients and their response to therapies and drugs can be extremely unpredictable.

The medical sciences generate an ever increasing amount of medical evidence and clinical variables potentially able to play a key role in controlling diseases or in predicting toxicities. For this reasons, due to the rising number

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

of covariates but the limited number of patients that can be enrolled for each hospital or research center, the trend is to move towards multi-centric clinical trials [7, 6]. A PM investigation exploiting multi-centric data sources would be able to extract more evidence and trends, to highlight organizational differences between participating centers, and to show different ways to cope with daily clinical and administrative issues. The perspective of Process Mining applied on different institutional data sources gives rise to problems related to data ownership and patients' privacy. The former concerns the need to merge all the data before beginning the investigation: which is something that can, for instance, put a fraudulent data manager in the position to maliciously exploit his or her role; the latter concerns one of the most pivotal constraint in managing clinical data, which is to cope with heterogeneous laws and different, country-dependent requirements. The best way to avoid any privacy-related issue, while still exploiting multi-centric data, is to avoid sharing clinical data among different hospitals, and perform analysis and training of algorithms by sharing only few aggregate parameters among hospitals and guaranteeing the global convergence (consensus) to an acceptable shared model. This paradigm is commonly termed distributed learning [8, 5]. While a number of approaches have been proposed to perform distributed machine learning, there is a lack of approaches for performing distributed multi-centric PM, where patients' privacy and data ownership are ensured and protected.

In this paper, we introduce the idea of distributed process mining, by focusing on one of the most well-known algorithms for process discovery: the Alpha algorithm [1]. This algorithm represents a milestone in the history of process discovery algorithms. Due to its simplicity it is widely adopted for describing the basic dynamics of process discovery, and it provides an invaluable ground for describing how distributed process mining can be performed.

2 Alpha Algorithm

The Alpha algorithm (AA) is one of the most well-known algorithms for process discovery [1]. It is widely appreciated for its simplicity, and it provides a very good ground for understanding how to move the first steps in process discovery.

A key concept on which the AA is based is the idea of the Footprint Matrix (FM), which is a squared matrix containing the ordering relations between pairs of events in the event log traces. In order to build this matrix, the following Log-Based ordering relations have to be defined. Given an event log L composed by a set of traces $L = \{\sigma_i\}$, the relation $>$ allows to express if a symbol b follows another symbol a: it is then possible to define that $a > b$ if, in at least one trace in L, in the list of terms t, a appears before b. Formally:

$$a > b \Leftrightarrow \exists \sigma_i = (t_1, t_2, .., t_n) \wedge j \in \{1, .., n-1\} \Rightarrow t_j = a \wedge t_{j+1} = b \qquad (1)$$

Using $>$ it is possible to define three other relations: $\rightarrow$, $\sharp$ and $\|$, as:

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

Algorithm 1 The Alpha algorithm, given an event log L and a set of traces σ.

1: $T_L = \{t \in T / \exists_{\sigma \in L} t \epsilon \sigma\}$
2: $T_I = \{t \in T / \exists_{\sigma \in L} t = first(\sigma)\}$
3: $T_O = \{t \in T / \exists_{\sigma \in L} t = last(\sigma)\}$
4: $X_L = \{(A, B) / A \subseteq T_L \cap A \neq \emptyset \wedge B \subseteq T_L \wedge A \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_L b \wedge \forall_{a1,a2 \in A} a1 \sharp_L a2 \wedge \forall_{b1,b2 \in B} b1 \sharp_L b2\}$
5: $Y_L = \{(A, B) \in X_L / \forall_{(A',B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$
6: $P_L = \{p_{(A,B)} / (A, B) \in Y_L\} \cup \{i_L, o_L\}$
7: $F_L = \{(a, p_{(A,B)}) / A, B \in Y \wedge a \in A\} \cup \{(p_{(A,B)}, b) / (A, B) \in Y \wedge b \in B\} \cup \{(t, o_L) / t \in T_O\}$
8: $\alpha(L) = (P_L, T_L, F_L)$

$$a \rightarrow b \Leftrightarrow a > b \wedge NOT(b > a) \tag{2}$$

$$a \sharp b \Leftrightarrow NOT(a > b) \wedge NOT(b > a) \tag{3}$$

$$a \| b \Leftrightarrow a > b \wedge b > a \tag{4}$$

For example, given the event log $L = \{< a, b, c, d >, < a, c, b, d >, < b, c >\}$, the following ordering relations can be specified:

- $a \rightarrow b$, because in the first trace b follows a but the opposite never happens (in the second trace a is not after b).
- $b \| c$, because $b \rightarrow c$ in the first and the third traces but $b \leftarrow c$ in the second.
- $a \sharp d$ because a and d never appear next to each other.

Given the specified ordering relations, the FM can then be built for the considered event log L. For the considered example, the FM would look as follows:

	a	b	c	d
a	$\sharp$	$\rightarrow$	$\rightarrow$	$\sharp$
b	$\leftarrow$	$\sharp$	$\|\|$	$\rightarrow$
c	$\leftarrow$	$\|\|$	$\sharp$	$\rightarrow$
d	$\sharp$	$\leftarrow$	$\leftarrow$	$\sharp$

The Alpha algorithm is summarized in eight steps, shown in Algorithm 1.

For a detailed description of the algorithm, the interested reader is referred to [2]. For the sake of conciseness, here we provide an overview of the main steps of the Alpha algorithm. In a nutshell, the steps indicated in Algorithm 1 aims: (1) to build the set of symbols adopted in the traces; (2) and (3) to build, respectively, the initial and final transitions; (4) by considering the Footprint Matrix, it is possible to find the groups of symbols which are in a $\sharp$ relation among them and in a $\leftarrow$ relation with all the subsequences (which, again, are in a $\sharp$ relation among them); (5) to reduce the set of the previous set of groups and build the set of states; (6) to define the arcs between couples of items identified

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

in the previous step; (7) to join the arcs with the states; (8) to compose the result and return the structure of the Petri net. The Petri net summarizes the process mined by the algorithm, and can then be exploited for the PM purposes. It should be noted, however, that states in the Alpha algorithm are mapped into nodes of the Petri net. While this can be confusing, this is due to the fact that the notion of state of a Petri net refers to the position of the markings. The interested reader is referred to [9] for an extensive description of Petri nets.

It can be observed that the Alpha algorithm needs only a multi-set of traces comprising exclusively activity labels: the information about users, the data artifacts such as clinical files, or the indications about the premises where processes are carried out, are completely unnecessary to the Alpha algorithm. In such a setting, the reader may object that privacy would not be violated by freely circulating traces (maybe pre-filtered in order to show only literals encoding activity labels) and subsequently join them. At a closer look, however, it becomes apparent that a traditional man-in-the-middle attack can violate the privacy of patients. If the attacker has knowledge (even partial) about the way in which activity labels are encoded, traces can be quickly converted into the corresponding ordered sequence of clinical activities. At this point, patients can be matched with sequences in cases where even a small subsequence of activities is known to the attacker, effectively allowing the attacker to have a complete overview of the health record of the patient.

3 Distributed Alpha Algorithm

It is easy to observe that the Alpha algorithm only requires the FM, T_L, T_I, and T_O sets to perform the process discovery task. Therefore, the event logs do not need to be considered and shared during the computation.

The FM represents an aggregation of the original data, that is generally effectively hiding single patient's pathways. In a canonical non-privacy-preserving multi-centric study, a data manager is required to merge the different event logs L_i collected from each of the participating center i, into a single event log L, and builds a single Footprint Matrix $FM(L)$. The Alpha algorithm can then be applied to compute the global Petri net PN. Here we introduce how to distribute the Alpha algorithm computation.

Each center locally computes its own $FM(L_i)$ and sends it, instead of the event log L_i, to the master node. In this way, the master can only see an aggregation of the data, and has no access to the individual traces. Subsequently, the master *adds* the $FM(L_i)$ by a relation $\overline{\overline{\oplus}}$ in order to obtain:

$$FM(L_1)\overline{\overline{\oplus}}FM(L_2)\overline{\overline{\oplus}}...\overline{\overline{\oplus}}FM(L_n) = FM(L) \tag{5}$$

At this point, having $FM(L)$, the master node can compute the Alpha algorithm following the traditional approach shown in Algorithm 1. It is therefore pivotal to define a relation $\overline{\overline{\oplus}}$ satisfying the property:

$$FM(L_a \cup L_b) = FM(L_a)\overline{\overline{\oplus}}FM(L_b) \tag{6}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

To understand how this relation should work, in the following we will analyze the behavior of each of the relations involved in the Footprint matrix, with particular regards to matrices composition.

Relation $\rightarrow$: The relation $a \rightarrow b$, requires that the symbol a appears at least once immediately before b, and that b never happens immediately before a. It is easy to show that this relation is associative with respect to the composition of event logs. Given two event logs L_j and L_k, if $a \rightarrow_{t1} b$ in L_j and $a \rightarrow_{t2} b$ in L_k, it follows that $a \rightarrow_{t1 \cup t2} b$ in $L_j \bigcup L_k$, because the Condition 2 is preserved. Again, (b) if $a \rightarrow_{t1} b$ and $a \sharp_{t1} b$ the condition holds, because it requires *at least* one occurrence. In the case (c) of $a \rightarrow_{t1} b$ and $a \|_{t1} b$ the situation is different, because $a \|_{t1} b$ introduces an occurrence of $b > a$, so the new relation between a and b becomes $a \|_{t1} b$ (see Equation 4). The last situation is $a \rightarrow_{t1} b$ and $a \leftarrow_{t2} b$ where evidently the result is $a \|_{t1} b$, because of 4. The case of $\leftarrow$ is trivial.

Relation $\sharp$. The relation $a \sharp b$ requires (see Equation 3) that a and b never occur adjacently in the traces. This means that in composing $\sharp$ with $\leftarrow$, $\rightarrow$ or $\|$, the result becomes respectively $\leftarrow$, $\rightarrow$ or $\|$.

Relation $\|$. Once two symbols are in relation of $\|$ for L_j, the new evidence of the existence of a $\leftarrow$ or $\rightarrow$ relation in L_k, does not change the more general relation $\|$ in $L_j \bigcup L_k$ between the two symbols. The same holds for $\sharp$, so the composition of $\|$ and $\sharp$ returns $\|$.

Missing Symbols. It is expected that not all the symbols in L_j appear also in L_k, and *vice versa*. In this case the only definition which holds is 2, and the corresponding symbols can be paired with $\sharp$.

The relations $\oplus$ and $\overline{\overline{\oplus}}$ We can define the relation $\oplus$ to join the contribute of different event logs L_j and L_k according to the following rules:

$\oplus$	$\sharp$	$\rightarrow$	$\leftarrow$	$\|$
$\sharp$	$\sharp$	$\rightarrow$	$\leftarrow$	$\|$
$\rightarrow$	$\rightarrow$	$\rightarrow$	$\|$	$\|$
$\leftarrow$	$\leftarrow$	$\|$	$\leftarrow$	$\|$
$\|$	$\|$	$\|$	$\|$	$\|$

In particular: $\sharp$ is a *neutral element* for the relation $\oplus$; $\|$ is an *absorbing element* for the relation $\oplus$.

Because the relation $\oplus$ works only with pairs of symbols, to operate with Footprint matrices we can define $\overline{\overline{\oplus}}$ as:

$$FM(L_T) = FM(L_A) \overline{\overline{\oplus}} FM(L_B)$$

where the i, j element of $FM(L_T)$ is equal to $FM(L_A)_{i,j} \oplus FM(L_B)_{i,j}$ and $FM(L_A)$ and $FM(L_B)$ have the same symbols ordered in the same rows/columns. In other words, $\overline{\overline{\oplus}}$ operates on matrices applying $\oplus$ to each corresponding element of the two given input matrices. In the case of missing values, for example if $FM(L_B)$ does not have the symbol z which appear in $FM(L_A)$, the column and the row of z should be added in $FM(L_B)$ in order to calculate $FM'(L_B)$ which have, in the same position of $FM(L_A)$, the missing event related to the other with $\sharp$.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

Example 1. In order to give a practical example, let us consider the case of two event logs: $L_j = \{< a, b, c, d >, < a, c, b, d >\}$ *and* $L_k = \{< b, c >\}$*. Starting from the two logs, we are interested in generating the FM of* $L = L_{j \cup k}$*.*

By applying the relations defined in Equations 2, 3, and 4, for L_j *and* L_k*, the corresponding Footprint matrices* $FM(L_j)$ *and* $FM(L_k)$ *would be, respectively:*

$FM(L_j)$

	a	b	c	d
a	♯	→	→	♯
b	←	♯	‖	→
c	←	‖	♯	→
d	♯	♯	←	←

$FM(L_k)$

	b	c
b	♯	→
c	←	♯

It is easy to notice that $FM(L_k)$ *is smaller then* $FM(L_j)$*, due to missing symbols in the corresponding event log* L_k*. In order to obtain matrices of the same shape and size,* $FM'(L_k)$ *has to be calculated, by filling the rows and columns corresponding to missing symbols with the* ♯ *relation, following the discussion provided in Section 3. As the matrix* $FM(L_j)$ *already includes all the symbols contained in any of the event logs, there is no need to manipulate it. It is now possible to calculate* $FM = FM(L_j) \overline{\overline{\oplus}} FM'(L_k)$*. The resulting matrix would then be as follows.*

$FM(L_j)$

	a	b	c	d
a	♯	→	→	♯
b	←	♯	‖	→
c	←	‖	♯	→
d	♯	←	←	♯

$\overline{\overline{\oplus}}$

$FM'(L_k)$

	a	b	c	d
a	♯	♯	♯	♯
b	♯	♯	→	♯
c	♯	←	♯	♯
d	♯	♯	♯	♯

=

$FM(L)$

	a	b	c	d
a	♯	→	→	♯
b	←	♯	‖	→
c	←	‖	♯	→
d	♯	←	←	♯

■

Having defined how different FMs can be composed via the appropriate relation, we are now in the position to describe the distributed Alpha algorithm. The distributed Alpha algorithm, inspired by the distributed paradigm proposed by Boyd et al. [4], includes the following steps:

1. Each center calculates the local Footprint matrix FM_{L_i} and the sets T_{L_i}, T_{I_i}, T_{O_i} (as discussed in Section 2);
2. the generated structures $< FM_{L_i}, T_{L_i}, T_{I_i}, T_{O_i} >$ are sent to the master;
3. the master node builds the sets:
   - $T_L = \bigcup_{i=1,..,n} T_{L_i}$
   - $T_I = \bigcup_{i=1,..,n} T_{I_i}$
   - $T_O = \bigcup_{i=1,..,n} T_{O_i}$
4. starting from FM_{L_i}, the master node calculates the corresponding FM'_{L_i}. This is the correctly ordered FM, where missing columns/rows –corresponding to missing events in a local event log– are filled with ♯, according to what has been shown in the corresponding section.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36

5. the overall FM is generated by composing all the FM'_{L_i} via the relation $\overline{\overline{\oplus}}$;
6. the master applies the Alpha algorithm on the structure $< FM, T_L, T_I, T_O >$;

This flow ensures the patient's privacy preservation and allows to apply the Alpha algorithm considering the entire set of available traces.

4 Conclusion

The complex causal relations between involved variables and values, and the objective difficulty of enrolling large cohorts of patients for a single medical center, lead to a growing need to merge data from different centers and perform multi-centric clinical analysis. A PM investigation exploiting multi-centric data sources would be in the best position to highlight organizational differences between different centers, and to show different ways to cope with daily clinical and administrative issues.

In this work, we introduced the first approach to perform multi-centric process mining. Building on top of the well-known Alpha algorithm, we designed a privacy-preserving technique that allows to perform distributed process mining while preserving patients' data privacy. The proposed approach is guaranteed to converge to the same model that would have been generated by merging all the different data sets, and has been empirically tested on a large set of event logs. Future work will focus on extending our approach to different algorithms, and to investigate different aspects of process mining, such as conformance checking and enhancements.

References

1. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. vol. 16, pp. 1128–1142 (2004)
2. van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag (2011)
3. van der Aalst W.M.P., et al: Process mining manifesto. In: Business Process Management Workshops. pp. 169–194 (2011)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1122 (2011)
5. Damiani, A., Vallati, M., Gatta, R., Dinapoli, N., Jochems, A., Leist, D., van Soest, J., Dekker, A., Valentini, V.: Distributed learning to protect privacy in multi-centric clinical studiest. In: Artificial Intelligence In Medicine (2015)
6. George, M., Selvarajan, S., Dkhar, S., Chandrasekaran, A.: Globalization of clinical trials - where are we heading? Curr Clin Pharmacol **8**(2), 115–23 (2013)
7. Gresham, G., Ehrhardt, S., Meinert, J., Appel, L., Meinert, C.: Characteristics and trends of clinical trials funded by the national institutes of health between 2005 and 2015. Clin Trials. **15**(1), 65–74 (2018)
8. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Proceedings of CRYPTO. pp. 36–54 (2000)
9. Peterson, J.L.: Petri net theory and the modeling of systems (1981)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_36