Learning the Context of a Clinical Process

Johny Ghattas[1], Mor Peleg[1,2], Pnina Soffer[1] and Yaron Denekamp[3]

[1]Department of Management Information Systems, University of Haifa, Israel, 31905
[2]Center of Biomedical Informatics, Stanford University, Stanford, CA, 94305
[3]Carmel Medical Center, Haifa, Israel
GhattasJohny@gmail.com, morpeleg, pnina @{mis.hevra.haifa.ac.il},
Yarondp@clalit.org.il

Abstract. Clinical guidelines provide recommendations to assist clinicians in making decisions regarding appropriate medical care for specific patient situations. However, characterizing these situations is difficult as it requires taking into account all the variations that patients may present. We propose an approach which helps with identifying and categorizing the contexts that need to be taken into account within a clinical process. Our methodology is based on a formal process model and on a collection of process execution instances. We apply machine-learning algorithms to group process instances by similarity of their paths and outcomes and derive the contextual properties of each group. We illustrate the application of our methodology to a urinary tract infection management process. Our approach yields promising results with high accuracy for some of the context groups that were identified.

Keywords: clinical guidelines, context, business process learning, process goals, soft-goals, process model adaptation, flexibility**.**

1 Introduction

Clinical guidelines are systematically developed statements to assist practitioner and patient decision making about appropriate health care for specific clinical circumstances [1]. They aim to improve patient care, limit unjustified treatment variation, and reduce costs. However, a clinical guideline cannot possibly address the variations in patient populations that occur in different healthcare institutions who try to apply the guideline. For example, the guideline may recommend that a certain conventional antibiotic should be given to patients with urinary tract infection (UTI) but that for patients who are resistant to the antibiotic (i.e., the pathogens which caused UTI in the patient are resistant to the antibiotic), a different antibiotic should be provided. Since giving a patient a non-effective treatment has many risks, in particular, that the patient's condition would deteriorate, the goal is to know under what context a patient is likely to be resistant to the conventional antibiotic. Guidelines often leave the conditions under which a patient is likely to be resistant to antibiotic undefined.

In this paper we propose to learn the different contexts relevant to UTI treatment in a local hospital, by mining electronic healthcare records (EHRs) of UTI patients.

To this end, we apply a context based process learning methodology which we have developed. We postulate that the context [2][3] of a process, namely, information about the properties and environmental events of each medical case, affect the process' execution and outcomes. However, the significant affecting variables and their effect are not necessarily known. Our approach aims at categorizing possible environmental conditions and case properties into context categories which are meaningful for the process execution. The context learning algorithm is part of a business process learning framework that we are developing, in which the best path would be proposed for each context group.

The remainder of the paper is structured as follows. Section 2 explains what a clinical context is and provides the motivation for the context learning framework. Section 3 describes our context learning framework and section 4 illustrates the application of context learning to a clinical process - the Urinary tract infection management process. Finally Section 5 discusses the implications of the model and compares it to models proposed in the past that have some similarity to our model.

2 Clinical contexts

Clinical processes (anamnesis, diagnosis, treatment) highly depend on the characteristics of each patient as well as on environmental conditions (e.g., availability of medical equipment and expertise in a healthcare facility).

Clearly, different contextual conditions should be handled by different paths for the process to achieve its goals. To facilitate this, three main challenges need to be met. First, normally there is no obvious way to establish a full repository of all possible context variations that are yet to appear. Second, while it is possible to have information about an (almost) unlimited amount of case properties, we should be able to identify which specific properties have an effect on the process. Third, medical organizations need to know how to select their process paths per each one of these situations in order to achieve the best outcome.

In this work, we focus on the second challenge and demonstrate via a case study of a UTI process the application of a context-learning framework that we developed. Context groups cluster together process instances that at the same time have similar path and outcome and can be grouped by sets of shared contextual properties, thereby limiting the number of context variations to be dealt with. This can be a first step towards defining process paths for each context group, such that taking that path would lead to desired process outcomes (goals). For this purpose, we target an active process, namely, a process which has already been executed for a while, and acquired past execution data. Our basic assumption is that in these past executions, some cases were addressed “properly” according to their relevant contextual properties (although a relation between context and path selection was not necessarily formally specified). Other cases were not properly addressed, and this should be reflected in the performance achieved by the process for these cases, which should be lower when compared to the properly addressed cases. Hence, the proposed methodology is based on clustering process instance data of past executions, relating to their context, path, and outcomes.

3 An approach for learning context groups of business processes

In this section, we briefly describe our context learning framework [11].

A business process instance (PI) is completely defined given its context, path, and attained termination state (goal or exception). In addition, our initial knowledge of the business process model provides us with the criteria necessary to identify when we reach our goal or when the process terminates in an exception. The information of the business process model and its execution data enable us to learn the relevant context groups. In this paper, we consider only goal states as outcomes, and not exceptions.

However, it is not uncommon *not* to have a completely defined process model or complete contextual information of the recorded process instances. Our context learning approach can use partial knowledge of the process model and approximate similarity measurements of different process instances.

Rather than estimating the similarity of paths separately from the similarity of outcomes, we apply machine learning algorithms to existing path and goal state data together. We developed a clustering strategy constructed of the following stages.

(1) We partition our process instances based on existing domain knowledge. For example, existing UTI guidelines partition patients into populations that depend on age, gender, catheter usage, etc.
(2) We estimate the similarity between all process instance paths and goal states, establishing a measure of the similarity of these instances, and grouping them into clusters. Technically, this is done by representing the process instance path and outcome data as vectors of values of state variables and using a clustering algorithm, to find clusters based on vector proximity. Then we use feature selection to omit state variables that are not important for determining the clusters.
(3) Once the clusters of PIs (PICs) have been identified, we apply supervised-learning algorithms (algorithms that build decision trees and prune them) to determine the meaning of the context groups that correspond to these groups. To do so, we focus only on the context information of the PIs of each PIC and express the meaning of the corresponding context group as a logical condition over the set of context variables. The context groups learning procedure is schematically shown in Figure 1.

4 Context group learning in urinary tract infection (UTI) management process

We apply the context learning framework to a clinical process dealing with urinary tract infection (UTI) patients.

4.1 UTI- a brief overview

Different healthcare organizations have developed their own guidelines for diagnosing and managing UTI. These guidelines indicate different care paths for

different partitions of the population, partitioned by age, gender, and other conditions, including the use of catheters and existence of complications related to arterial, heart and kidney diseases, and diabetes mellitus. The most important partition is the one concerning elderly women, which constitute more than two thirds of the impacted population.

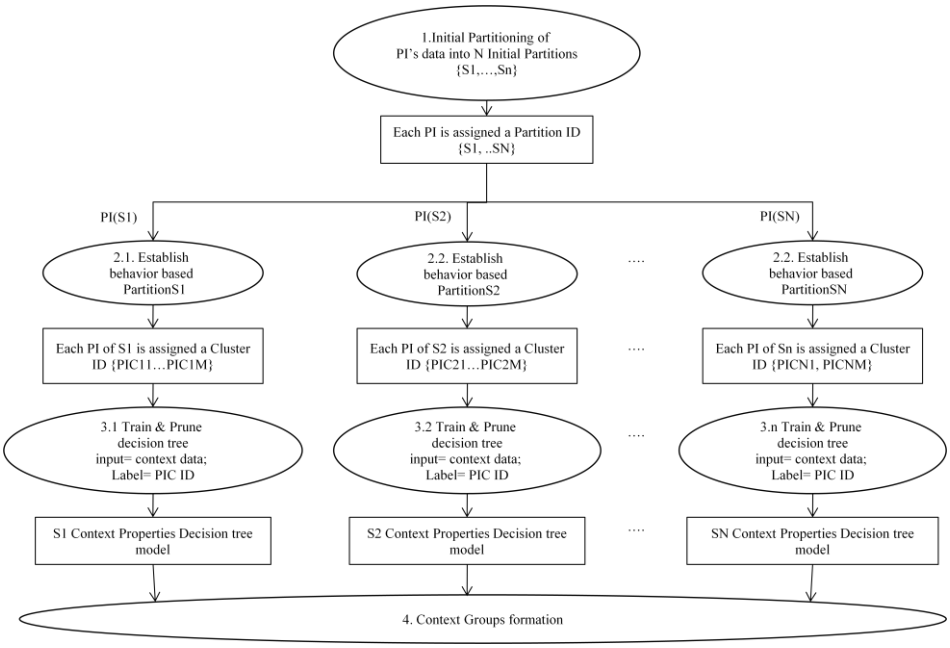

Figure 1. Architecture of the context groups learning algorithm. PI- process instance; S- Initial partition.

4.2 UTI process instances and context data

The data for our case study was collected at a general internal medicine department in Carmel Medical Center, Haifa, Israel, and includes 297 patient records. Most patients in our database are elderly persons (above 50 years old), who arrived first at the emergency room, where they were diagnosed as potentially having UTI. Then they were admitted into the general internal medicine department. Most of the context data is known from the medical record of the patient (either electronically (EHR) or paper-based) and is further collected from the patient as a first step of the process. This step is known as "anamnesis". In it, the physician questions the patient to identify chronic illnesses, medications that the patient is taking for other conditions, symptoms he is having, whether UTI is a recurring problem, historical illnesses related to UTI (such as calculi existence, reflux problems, kidney problems, etc.), general test results (urinalysis), and physical examination results. A partial list of context data includes: (1) age, (2) gender, (3) race, (4) vital signs, such as fever, blood pressure, and heart rate, (5) symptoms, (6) physical examination results, (7) chronic illnesses, such as diabetes mellitus (DM), hypertension (HTN), coronary arterial disease (CAD), congestive heart failure (CHF), cancer/hematological disorder, chronic pulmonary

disease (CPD), chronic renal failure (CRF), cerebro-vascular disease (CVD), (8) medications, such as beta-blockers (BB), (9) previous UTI, (10) existence of a permanent catheter, (11) general mental and overall state of the patient, (12) whether UTI was acquired in the hospital, and (13) residence (e.g., nursing home).
Following the anamnesis and physical examination, the patient is diagnosed. Several diagnoses may be given and registered in the medical record; we consider up to ten different diagnoses, which impact the further diagnosis and treatment of UTI, including, among others, fever, hypertension, chronic renal failure, depression, anxiety, pneumonia.

Following the initial diagnosis, initial treatment may be provided (e.g., antibiotics or other medications) and additional tests may be ordered to further diagnose the patient's condition and evaluate the expected outcomes (prognosis). Tests may include urine culture tests, ultrasound, prostate examination for men, etc. The tests depend on the patient's context. The test results may arrive several days after the patient has been initially diagnosed and has undergone initial treatment. After the test results become available, the treatment may be changed and additional tests may be ordered.

Hence, the main activities in the UTI management process path and the main outcome state variables that we expect to be reflected in the patients' records include the following 6 data items: (1) the ten diagnosis terms (mentioned above), (2) initial treatment (with 27 kinds of antibiotics), (3) three categories of medical tests (urine culture, blood tests, ultrasound), (4) modified treatment (after test results return), (5) additional tests ordered after treatment has been modified (three possible tests), and (6) four possibilities of final status: death, cured, partially cured, follow up needed by other specialists. A partial sample of path data is provided in Table 1.

Table 1. Path data structure.

Process instance ID	253467
Initial Treatment	< Augmentin>
Diagnosis	<CVD, CRF, UTI>
Urine Culture test results	<…>(1 field for each measure), <ESBL+= Y>
Blood test results	<…> (1 field for each measure)
Ultra sound	<OK>
Modified treatment	< ZINACEF>
Additional tests	<<CT, OK>, <ESBL+>
Final Patient status	<Partially cured- require home care >

4.3 Establishing context groups for the UTI data

Different patients may have different initial conditions, such as different symptoms and different chronic illnesses. Hence, the UTI diagnosis and treatment process may vary from one patient to another. The question we are trying to answer using our context learning framework is: can we group patients' data into context groups in such a way that consistent outcomes are achieved for a defined set of process paths for each group?

We describe how our context learning algorithm follows the three steps defined in Section 3 for the UTI case.

Step 1: Initial partitioning of context data. We partition the data based on different populations addressed in UTI clinical guidelines. Through a review of existing guidelines, we saw that UTI guidelines distinguish between the following partitions: (1) New born; (2) Children; (3) Pregnant women; (4) Young women; and (5) Elderly Men and women. Some guidelines distinguish between patients with permanent catheters and without catheters. Since most of the patients in our database are elderly men and women (above 50 years old), we will focus on analyzing partition #5.

Step 2: For the 297 patients, we recoded the process activities (e.g., medications, tests, procedures, diagnosis) and outcome state variables discussed in Section 4.2. Using a modification of the two-step clustering offered as part of the SPSS package [4], we clustered process instances (PIs) according to similar path and outcome data and assigned each PI to a PIC ID. To find a set of clusters that achieves good clustering results, we generated 15 cluster sets, consisting of 1 to 15 clusters, respectively. Using the Akaike information criterion (AIC) [5] as a measure of the goodness of fit of an estimated statistical model and is grounded in the concept of entropy, we identified the set of clusters that achieves the best results. The best cluster set partitioned the 297 samples into five clusters (PIC1 through ...PIC5) of size 54, 27, 51, 80, and 85 samples, respectively.

After the process instance data was clustered, we used the chi-square statistical test as a method for feature selection. Using the chi-square statistic, we analyzed the significance of each variable to each one of the five clusters in order to omit from the cluster features that are non-significant. For example, the variable “Urinary Cancer” is most significant for Cluster #3 but could be omitted from the context variables of the rest of the clusters. The variable “Renal Failure not including CRF” is highly significant for Cluster #2 and #3, but could be omitted from the context variables of clusters #1, #4, #5. Performing feature selection for each one of the variables reduced the number of variables representing the clusters' context by an average of 20 %.

Step 3: We partitioned the context variables of the PIs into 35 variables, categorizing the values of each variable into discrete ranges of values that would be significant for a medical expert. For example, the age was partitioned into the following ranges: 45-55, 55-65, 65-75, 75-85, 85-90 and 95-105 years. Based on the context data of the PIs clustered in each cluster, we used a modified Chi-squared Automatic Interaction Detection (CHAID) growing decision tree algorithm [6] to construct the decision tree that represents the context groups and their relationships (see Figure 2). We provide CHAID with the context data of the PIs and with the PIC ID of each PI, which was deduced in step 2 according to the path and outcome data of the PIs. The PIC ID serves as the dependent label. CHAID tries to split the context part of the PI data into nodes that contain PIs that have the same value of the dependent variable (i.e., which were labeled in step 2 by the same PIC ID). The root of the tree shown in Figure 2 is partition #5 (Male and Female patients over 50 years), selected in Step 1. From there, the tree-building algorithm hierarchically partitions the nodes further, using at each split a context variable that is most important for segmenting the tree node, importance being estimated by chi-square. For example, nodes 0 is split based on age. The semantics of the nodes are criteria over the state variables. Node 4, for example, corresponds to age in the range 45-55.

Although CHAID aims to split the root node into *clean* nodes, each containing PIs that received a single PIC ID label in step 2, not all nodes formed are clean. For example, in Figure 2, nodes 21 and 23 are clean, containing PIs that were labeled as PIC #4 and PIC #3, respectively, as seen by the single column in the bar graph for these nodes. On the other hand, node 8 is less clean than nodes 21 and 23 as it contains similar levels of PIs with different labels and hence it is hard to select the most probable value for this node when trying to classify instances through it. Therefore, we state that this node has a high level of prediction error, while nodes 21 and 23 have very low prediction error. The predicted PIC ID for each node in the tree is the PIC ID that minimizes the prediction error. A common way of minimizing the prediction error is choosing the most dominant PIC ID for the node. For instance, considering node 17 or node 21, the output would be PIC = 4 with a probability of 98%, for node 23 it would be PIC =3 with a probability of almost 100%, while for node 8 it is not possible to predict the value of the output.

We used a cross-validation procedure [13] to find the misclassification (prediction) error we may expect for future PIs that we would classify with the tree. Cross validation divides the sample into a number of subsamples, or folds. Tree models are then generated, excluding the data from each subsample in turn. The first tree is based on all of the cases except those in the first sample fold, the second tree is based on all of the cases except those in the second sample fold, and so on. For each tree, misclassification risk is estimated by applying the tree to the subsample excluded in generating it. Cross validation produces a single, final tree model. The cross-validated risk estimate for the final tree is calculated as the average of the risks for all of the trees. In Table 2, we cross-tabulate the actual PIC ID (column 1) that was used to train the tree with the predicted PIC ID of the final tree. For example, of the 54 PIs that were originally labeled in PIC #1, 37 were predicted by the tree to have label of PIC #1, but 7 PIs were predicted to have label #2, and 10 PIs to have label #5.

Our objective is that the tree model would provide the predicted PIC for every new PI that we would submit to it. The ideal case would be that each leaf node of the tree would contain instances from one single PIC. However, this is not feasible due to the inherent errors of machine learning classification, and in addition, due to data completeness and correctness issues that arise despite our best to have the data validated and corrected. Therefore, we cannot be sure that we have all the context-related variables neither can we be sure that the data source is 100 % correct.

More importantly, we assume that when the analysis is performed there is no definition of path per context group. So we cannot expect all instances of the same context to follow the same path; the process is performed differently for different instances, even if they belong to the same context group, simply because there are no defined decision rules that relate path to context. Therefore we are not expecting our learning approach to find perfect correspondence between context groups and PICs.

Moreover, it is very likely to see different levels of success, measured via the classification error ratio, for different clusters, as seen in Table 2. For example, we see that for PIC ID #1 and #2 we have less than 70% successful prediction rate, for PIC ID #3 and #4 we have a prediction rate of 74-78%, and for PIC #5 we have a success rate of over 90%. The overall classification success rate for the provided set of data is 72%.

Table 2. Tree cross-validations results for the UTI process, considering the elderly males and females partition.

PIC ID used to train the tree	PIC ID predicted by the decision-tree					
	1	2	3	4	5	% Correct
1	37	7	0	0	10	67.7%
2	13	13	20	0	5	39.7%
3	1	21	0	0	5	78.2%
4	11	9	0	60	0	74.6%
5	7	1	0	1	76	90.0%
Overall %	23.3%	17.1%	6.9%	20.5%	32.2%	72.0%

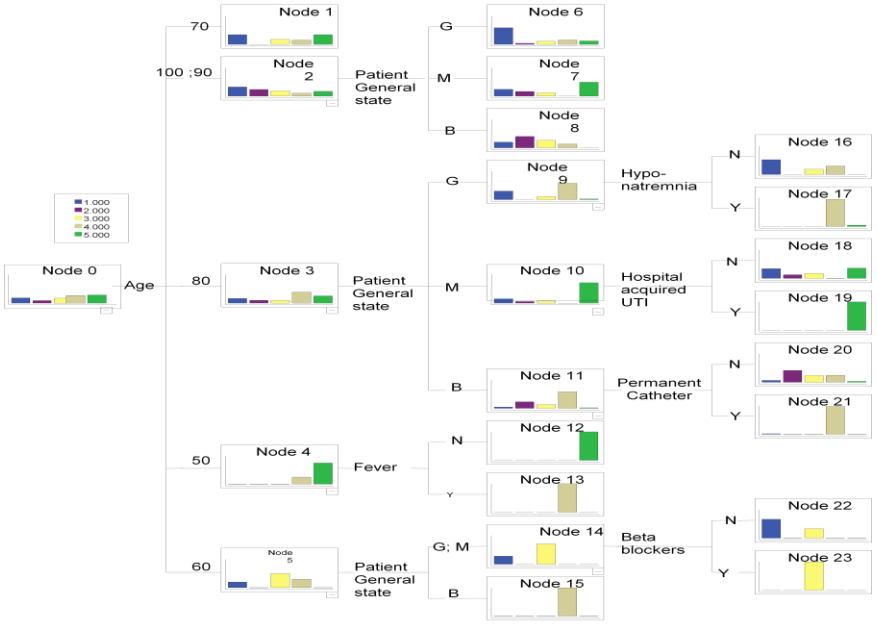

Figure 2. Decision tree resulting from applying Step 3 to the UTI process data. The initial node (Node 0 at the left) is the starting point of the process of growing the tree, corresponding to partition 5 obtained in Step 1. The variables that are used to split n nodes are written to the right of each node. The thresholds of the variable that determines the criteria representing each node are marked at each split. For example, Node 0 is split over the variable age into 6 different splits (50, 60, 70, 80, 90-100), indicating the age ranges 45-55, 55-65, 65-75, 75-85, 85-90 and 95-105. Patient_General_state has values Bad (B), Medium (M), and Good (G). The other variables are Boolean. The histogram shown at each tree node represents the number of vectors in the tree node that were labeled with a specific label.

4.4 Identifying context groups

Once we have built the decision tree, we define the context groups' logical conditions using the following steps. First, we label the tree's leaf nodes by walking through the tree from its root, collecting state variables and variable values used to split nodes. For example, tree node #23 is labeled as "55 <age < 65 AND (General_state = Medium or General_state = Good) AND Beta Blockers= Y". In this way we label the other 14 leaf nodes (1, 6, 7, 8, 10, 12, 13, 15-23).

Then, we examine the population of labels for each node; the different colors given for a single tree node represent the five PIC labels that were used to train the tree. The histogram shown at each tree node represents the number of vectors in the tree node that were labeled with a specific label. For example, the PIs in tree node #23 are all labeled with PIC #3, whereas the PIs in tree node #6 are labeled mostly with PIC #1. Of the 15 tree nodes, 9 are predominantly labeled by a single PIC. We use them to determine the logical condition that defines the PIC. The results are shown in Table 3.

As a measure of *sensitivity* [14], we calculated how many (percentage) of PIs that belonged to a given PIC also belonged to the tree nodes in the CG from which the semantic label for the CG was derived. As a measure of *specificity* [14], or "cleanness" of the semantic label, we calculated the number (percentage) of PIs in a tree node that received the predominant label for the node. We noticed that we had 3 categories (CGs 3, 4, and 5) that had specificity above 95%. These groups included 47.5% of the sample.

5 Discussion and Conclusions

In this paper we addressed the identification of context groups of a clinical process. A clinical process would be executed differently for different context groups. Hence, the identification of context groups helps in defining decision-support for clinical processes. In the medical informatics literature, ideas similar to context have been used for decision-support for clinical processes. Tu et al. [7] proposed the consideration of usage scenarios in order to identify opportunities for providing decision support, the roles and information needs of care providers, events that may activate the guideline system, and guideline knowledge relevant in these scenarios. The usage scenarios are derived by mapping of generic guidelines to specific medical institutions and drive the whole process of clinical process design by providing the process with all necessary inputs: “who is doing what, where and when”. A similar idea of context is also used in the definition of Act classes in Health Level 7's Reference Information Model (RIM) [8]. Taking an action-centered view, Act classes identify the kind of action (what happens), the actors who accomplish the action, the objects or targets whom the action influences. Adverbs of location (where), time (when), manner (how), and other information about circumstances, such as reasons (why) or motives (what for) are additional pieces of information that may be required or optional in given situations.

Process mining has been applied to healthcare processes [9]. The objective of process mining is to discover out of the process data the process model that has been

followed. In our work we depart from the assumption that the current business process model is known, or has been discovered though process mining, and we apply our framework to discover out of the process path and events the context of each instances. We consider that our context learning framework can be used by process mining algorithms in order to first establish groups of instances which are similar at the path and outcomes level and then discover the associated paths. This would provide process mining frameworks with two main capabilities: first, taking process outcomes into account when discovering the path, which, as we show, is an essential element for distinguishing between similar and non-similar instances; second, focusing the discovery on groups instead of mining all instances, which should improve notoriously the performance and the quality of the mining results.

Table 3. Resulting context groups for the UTI management process. The logical criteria are given for each relevant tree node. When a context group contains more than one tree node, the logical conditions of the nodes are combined with an OR to obtain the context group definition.

CG#	Tree Node	Logical Criterion	Sensitivity	Specificity	% of total sample
1	6	85 <age < 105 AND General state =Good	53.1%	58.6 %	6.6%
	22	55 < age < 65 AND (General state = Medium or General_state = Good) AND Beta- Blockers=”N”	46.9%	66.7%	5.1%
2	--				
3	23	55 <age < 65 AND General state = Medium or Good AND Beta Blockers= Y	100%	100%	6.9%
4	13	45 <age <55 and Fever =Y	22.7%	100%	4.6%
	15	55 <age <65 AND General state = Bad	25.6%	100%	5.1%
	17	75 <age <85 AND General state = Good AND Hyponatremia=Y	23.9%	95.8%	4.8%
	21	75 <age <85 AND General state = Bad AND Permanent Catheter=Y	27.8%	98.0%	5.6%
5	12	45 <age < 55 AND Fever=N	33.3%	100%	13.7%
	19	75 <age <85 AND General state = Medium AND hospital acquired UTI =Y	66.7%	100%	6.9 %
Total and weighted averages			45.5%	92%	59.2%

We have demonstrated the context learning framework by applying it to a clinical process in order to automatically deduce context groups. We postulate that the process path and outcomes are highly dependent on the process context, which specifies the inputs of the external environment to the process and hence constrains the adopted path and the reached termination state. Our approach is based on clustering similar process instances and then using the cluster IDs as labels for a

decision-tree learning algorithm from which semantic labels are extracted. The semantic labels are logical predicates over process state variables. This procedure renders the task of identification of contexts easier for a medical expert, enabling him to focus on analyzing the required paths for each context group without needing to deal with hundreds of samples. When a context group contains more than one tree node, we combined the logical conditions of the nodes with an *or* relation to obtain the context group definition. For each context group we will recommend one path. However, the different tree nodes that belong to the same context group are kept distinct, as they belong to different patient populations. It is important to keep them separated in this way so that the domain experts would relate to them clearly.

The resulting decision tree not only provides semantic labels for context groups; it may also be used to identify the context group of future instances automatically.

We note that our knowledge and definition of contexts is usually limited. Establishing a fully-accurate context definition would require having all the state variable data collected, which is impractical. We also need to expect some level of error in the provided data. All this, in addition to the inherent error of classification, which is in the nature of machine learning, implies that we always need to account for some level of classification error. Before applying our technique for deducing semantic labels from the clean nodes of the decision tree, the overall prediction level of the tree was 72% (Table 2) but it was not uniform. For example, in our study, the prediction level for PIC #2 was low - we did not have enough PIs in PIC #2 to learn a semantic definition for it. On the other hand, the results that we obtained for PIC # 3 were excellent – 100% specificity and sensitivity (Table 2 and Table 3). Comparing Table 2 and Table 3, we see that the method that we used for deducing the semantic labels produced high specificity (higher than the prediction rate observed in Table 2) because we used only clean nodes to provide the semantic labels but low sensitivity, because we dropped the PIs belonging to nodes that were not clean. These preliminary results, based on only 297 patients, are encouraging and show promise for our approach. We believe that when we collect more data, these results could be improved.

Our algorithm is a first component of a process learning architecture [2] that we have started to develop. The purpose of that approach is to learn, based on an initial process model schema and the outcomes of PIs, the process paths that should best be adopted for a PI that is awaiting execution. It is our goal to modify the initial process model schema based on the learned knowledge and achieve a better process model schema. Our approach differs from case-based reasoning (CBR) [10], which uses a case-base of PIs to propose for a given PI awaiting execution a similar PI from the case-base that achieved good outcomes. CBR has been applied to the domain of business process management [11].

Since our approach is generic and is based on a formal conceptual model definition of the process model, process context, and process outcomes [12], it could potentially be applicable to other domains. Future research directions would examine this prospect.

References

1. Field MJ, Lohr KN. Guidelines for Clinical Practice: Directions for a New Program. Washington DC: Institute of Medicine, National Academy Press; 1990.
2. Ghattas J, Soffer P, Peleg M. A Goal-based approach for business process learning. Workshop on Business Process Modeling, Development, and Support (BPMDS'08), in conjunction with CAISE'08; Montpellier, France; 2008.
3. Ploesser K, Peleg M, Soffer P, Rosemann M, Recker J. Learning from Context to Improve Business Processes. BPtrends 2009(1):1-9.
4. SPSS corporation. SPSS statistics software, version 16. In; 2009.www.spss.com.
5. Akaike H. A new look at the statistical model identification. IEEE Transactions on Automatic Control 1974;19(6):716-723.
6. Kass GV. An Exploratory Technique for Investigating Large Quantities of Categorical Data. Journal of Applied Statistics 1980;29(2):119-127.
7. Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, J McClay J PC, Hrabak KM, Berg D, Weida T, Mansfield JG, Musen MA, Abarbanel RM. The SAGE Guideline Model: achievements and overview. J Am Med Inform Assoc 2007;14(5):589-98.
8. Russler DC, Schadow G, Mead C, Snyder T, Quade LM, McDonald CJ. Influences of the Unified Service Action Model on the HL7 Reference Information Model. Proc. AMIA Annual Symposium; 1999. p. 930-4.
9. Mans, R. S., Schonenberg, M. H., Song, M., Van der Aalst, W. M. P., Bakker, P. J. M. Application of Process Mining in Healthcare - A Case Study in a Dutch Hospital Biomedical Engineering Systems and Technologies, Communications in Computer and Information Science, Volume 25. Springer Berlin Heidelberg, 2009, p. 425
10. Aamodt A, E EP. Case based reasoning: foundational issues, methodological variations and system approaches. AI Communications 1994;7(1):39-59.
11. Weber B, Rinderle S, Wild W, Reichert M. CCBR-Driven Business Process Evolution. Proc. ICCBR'05; 2005. p. 610-24.
12. Ghattas J, Soffer P, Peleg M. A formal model for Process context learning. Accepted for pusblishing at BPI 2009, 5th workshop on Business process intelligence, Sept. 2009, Ulm, Germany.
13. Geisser, Seymour. Predictive Inference. New York: Chapman and Hall. 1993, ISBN 0412034719.
14. Simon D, Boring III JR. Sensitivity, Specificity, and Predictive Value. In: Clinical Methods: The History, Physical, and Laboratory Examinations, Walker HK, Hall WD, Hurst JW, eds. Butterworths, 3rd edition, 1990.