Detection of Sequences with Anomalous Behavior in a Workflow Process

Conference Paper · September 2015

DOI: 10.1007/978-3-319-22849-5_8

2 authors:

Marcelo G. Armentano
National Scientific and Technical Research Council
52 PUBLICATIONS 397 CITATIONS
SEE PROFILE

Analía Amandi
National Scientific and Technical Research Council
170 PUBLICATIONS 2,885 CITATIONS
SEE PROFILE

Detection of sequences with anomalous behavior in a workflow process

Marcelo G. Armentano and Analía A. Amandi

ISISTAN Research Institute (CONICET-UNICEN)
Campus Universitario, Paraje Arroyo Seco, Tandil, 7000, Argentina
{marcelo.armentano, analia.amandi}@isistan.unicen.edu.ar

Abstract. A workflow process consists of an organized and repeatable pattern of activities that are necessary to complete a task, within the dynamics of an organization. The automatic recognition of deviations from the expected behavior within the workflow of an organization is crucial to provide assistance to new employees to accomplish his/her tasks. In this article, we propose a two-fold approach to this problem. First, taking the process logs as an input, we automatically build a statistical model that captures regularities in the activities carried out by the employees. Second, this model is used to track the activities performed by the employees to detect deviations from the expected behavior, according to the normal workflow of the organization. An experimental evaluation with five processes logs, with different levels of noise, was conducted to determine the validity of our approach.

Keywords: process mining; outliers detection

1 Introduction

Every organization has a workflow dynamics that is usually not explicitly documented. All these skills and experience that is hold on the employees characterizes the "know how" of any organization and is called Organizational Memory. This workflow dynamics can be lost when employees with some specific knowledge leave the organization, a problem that is known as *corporate amnesia* [10]. For this reason it is desirable to have an explicitly representation of the workflow dynamics that can be transmitted to new employees. Automatically learning a model of the dynamics of an organization can be beneficial since the behavior of new employees can be compared with the normal behavior to detect deviations from it. In other words as a new employee performs a set of tasks, it would be desirable to non-intrusively detect, as early as possible, any deviation from expected behavior.

The increasing use of technology to align the business processes of an organization towards the same goal has made a strong trend towards process-oriented information systems, which have a whole infrastructure to support such business processes. Despite all the advantages obtained from the use of these systems, the continuing growth and dynamics of organizations make business processes grow in number and complexity, resulting in an increasing difficulty to support its design and its rapid adaptation to changes. To help overcome these problems, a research area known as *workflow mining [1]* has emerged. Taking data from the results of the execution of processes, which are derived from the execution logs, workflow mining is based on applying different data mining techniques to obtain additional knowledge such as building a new model of a given process in order to compare it to the original process, detecting deviations from the original process or improving the process definition itself.

In this context, the automatic recognition of the sequences of tasks that do not belong to the normal behavior according to the workflow of a give process is crucial to determine deviations from the expected behavior that might lead to an ineffective operation of the enterprise. An early recognition of this kind of outlier sequences can prevent deviations of the employees behavior from the expected behavior by providing personalized assistance. The main difference between the process mining approach and the approach proposed in this article is that we do not focus on obtaining an explicit design of the underlying workflow. Instead, we seek to obtain a model of the tasks involved in the underlying workflow in order to detect behaviors that do not fit the expected flow of activities. With this information available, a system will be able to provide personalized assistance in the execution of those tasks.

This article is organized as follows. Section 2 presents some background and related work on workflow mining and outlier detection in workflows. Section 3 presents the proposed approach to model workflow processes from activities logs and to detect sequences with abnormal behavior. Section 4 describes the experiments performed to validate our approach. Finally, in Section 5 we present our conclusions.

2 Related work

Cook and Wolf investigated the mining of processes in the context of software engineering processes. In [7] three methods for process discovery are described: one using neural networks, one using an algorithmic approach and the third using a Markovian approach, concluding that the latter two methods are more effective. Wen et. al [13] presented an algorithm based on two types of events that indicate the beginning and completion of tasks. Together with the causality information obtained from the activities log, they derive relationships between tasks which are then used to create a Petri net modeling the underlying process. Two disadvantages of this approach are that it is not probabilistic and that is not robust to the presence of noise in the training data. Two disadvantages of this approach are that it is not probabilistic and that it is not robust to the presence of noise in the training data.

Regarding the detection of outliers in workflow logs, Ghionna et al. [8] presented an algorithm that discovers a set of outliers, based on the computation of behavioral patterns over process logs and a clustering approach. The basic idea is to associate pattern clusters with trace clusters, and to detect as outliers those traces that do not associate to any pattern cluster or that belong to clusters whose size is smaller than the average cluster size. The research presented in [5] proposes an algorithm which makes use of the workflow's executed frequency, the concept of distance-based outlier detection, empirical rules and Method of Exhaustion to mine three types of workflow outliers, including less-occurring workflow outliers of each process, less-occurring workflow outliers of all processes and never-occurring workflow outliers. More recently, Bouarfa and Dankelman [3] derive a workflow consensus without prior knowledge using logs extracted from a clinical environment by aligning multiple sequences. This model is used to detect outliers during surgery, that is, deviations from the consensus.

In the following Section, we present our approach to obtaining a probabilistic model from execution logs. This model is then used to detect sequences with abnormal behavior.

3 Proposed Approach

3.1 Learning a workflow model from execution logs

Variable Order Markov (VOM) models arose as an alternative to fixed order Markov models to capture longer regularities while avoiding the size explosion caused by increasing the order of the model. In contrast to the Markov chain models, where each random variable in a sequence with a Markov property depends on a fixed number of random variables, in VOM models this number of conditioning random variables may vary based on the specific observed realization, known as *context*. These models consider that in realistic settings, there are certain realizations of states (represented by contexts) in which some past states are independent from the future states conducting to a great reduction in the number of model parameters.

Algorithms for learning VOM models over a finite alphabet Σ attempt to learn a subclass of Probabilistic Finite-state Automata (PFA) called Probabilistic Suffix Automata (PSA) which can model sequential data of considerable complexity. Formally, a PSA can be described as a 5-tuple$(Q, \Sigma, \tau, \gamma, \pi)$, where Q is a finite set of states, Σ is the task universe, $\tau : Q \times \Sigma \rightarrow Q$ is the transition function, $\gamma : Q \times \Sigma \rightarrow [0, 1]$ is the next task probability function, where for each $q \in Q$, $\sum\limits_{\sigma \in \Sigma} \gamma(q, \sigma) = 1$, $\pi : Q \rightarrow [0, 1]$ is the initial probability distribution over the starting states, with $\sum\limits_{\sigma \in \Sigma} \pi(q) = 1$.

A PFA is a PSA if the following property holds. Each state in a PSA M is labeled by a sequence of tasks with finite length in Σ^* and the set of sequences S labeling the states is suffix free. Σ is the domain task universe, that is the finite set of tasks that the employee can perform in the domain. A set of sequences S is said to be suffix free if $\forall s \in S$, $Suffix^*(s) \cap S = \{s\}$, where $Suffix^*(s) = \{s_i, \cdots, s_l | 1 \leq i \leq l\}$ is the set of all possible suffixes of s, including the empty sequence e. For every two states q_1 and $q_2 \in Q$ and for every task $\sigma \in \Sigma$, if

$\tau(q_1, \sigma) = q_2$ and q_1 is labeled by a sequence s_1, then q_2 is labeled by a sequence s_2 that is a suffix of $s_1 \cdot \sigma$.

In contrast to m-order Markov models, which attempt to estimate conditional distributions of the form $Pr(\sigma|s)$, with $s \in \Sigma^N$ and $\sigma \in \Sigma$, VOM algorithms learn such conditional distributions where context lengths $|s|$ vary in response to the available statistics in the training data. Thus, PSA models provide the means for capturing both large and small order Markov dependencies based on the observed data. In [2] an algorithm for learning such models in an incremental way is proposed.

Learning a workflow model from activities logs has the main advantage that we do not need any additional information about the domain being modeled more than the tasks that can be performed in the domain. We will be able to learn regularities in the employees' behavior just by analyzing the trace examples observed in the logs. In the following section we describe how we use PSA's models to detect sequences of abnormal behavior in execution logs.

3.2 Detecting sequences with abnormal behavior

In order to make the recognition process robust to the execution of noisy tasks, that are tasks that might not correspond to abnormal behavior, we use an *exponential moving average* on the prediction probability $\gamma(s, \sigma)$ at each step in the PSA. An exponential moving average (EMA) [9] is a statistic for monitoring a process that averages the data using weights that decrease as time passes. The weighting for each step decreases exponentially, giving much more importance to recent observations while still not discarding older observations entirely. By the choice of a weighting factor $0 \leq \lambda \leq 1$, the EMA control procedure can be made sensitive to a small or gradual drift in the process. Alternatively, λ may be expressed in terms of N time periods, where $\lambda = \frac{2}{N+1}$.

EMA_t expresses the value of the EMA at any time period t. EMA_1 is set to the a priori probability of the first observed task σ. EMA_t at time periods $t \geq 2$ is computed as $EMA_t = \lambda \gamma_{PSA_i}(s, \sigma) + (1 - \lambda) EMA_{t-1}$. The parameter λ determines the rate at which *older* probabilities enter into the calculation of the EMA statistic. A value of $\lambda = 1$ implies that only the most recent measurement influences the EMA. Thus, a large value of λ gives more weight to recent probabilities and less weight to older probabilities; a small value of λ gives more weight to older probabilities. The value of λ is usually set between 0.2 and 0.3 [9] although this choice is somewhat arbitrary and should be determined experimentally.

In our approach, we consider that a given sequence corresponds to an abnormal behavior if at least one activity in the log sequence is predicted by the PSA model with a probability value (after applying the smoothing technique) lower than a certain threshold. This is a very strong assumption that favors the detection of outliers over normal sequences.

	Process 1	Process 2	Process 3	Process 4	Process 5
Number of AND patterns	0	2	2	1	1
Number of XOR patterns	3	1	1	2	1
Number of loops	4	2	0	1	1
Maximum number of AND branches	0	3	2	3	2
Maximum number of XOR branches	3	3	2	3	2
Total number of activities	18	18	12	18	11

Table 1: Statistics for the processes used in the experiment

Sequence length	Process 1	Process 2	Process 3	Process 4	Process 5
Min.	7	14	11	6	6
Max.	293	66	11	92	33
Mean	43.78	23.16	11	26.18	10.36
Median	32	21	11	23	8
Standard deviation	37.28	8.79	0	14.98	5.34

Table 2: Statistics for the testing logs generated for the experiment

4 Experimental Evaluation

Process logs corresponding to real-world business process are hardly available. Companies owning real process logs are often reluctant to make public their data and release only partial log files. For these reasons, we have tested our technique with artificially generated process logs, as in [12] and [6]. The main benefit of using simulation is that properties such as noise can be controlled.

Different tools exist for generating artificial event logs. From the evaluation presented in [11], we selected PLG [4], a framework that enables the generation of random business processes according to some specific user-defined parameters. We defined five different workflows using PLG. Table 1 shows some statistics for the five process used in the experiment.

Next, for each workflow, we use PLG to generate a maximum of 500 traces and filtered out duplicate traces. Finally, we trained five different VOM models to be used in the experiments.

We evaluated the effectiveness of our approach with various input logs, containing different percentage of noise in the sequences. Using PLG, we generated three different logs for each process. Each log contained a maximum of 500 execution traces introducing 5%, 10% and 20% of noise in each sequence, representing abnormal behavior according to the workflow process. Each log was then completed with a maximum of 500 execution traces with no noise, representing normal behavior sequences[1]. Table 2 shows some statistics of the resulting logs

[1] The resulting dataset is available online at: http://marcelo.armentano.isistan.unicen.edu.ar/datasets

4.1 Selection of the smoothing constant λ

First, we tested different values for the smoothing constant, ranging from 0.1 to 1.0 with intervals of 0.1 for the five processes. In order to select the best λ value for each process, we computed the Matthews correlation coefficient (MCC). The MCC is in essence a correlation coefficient between the observed and predicted binary classifications and ranges between -1 and $+1$, where a coefficient of $+1$ represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation.

For processes 1, 3 and 4 we obtained better MMC with a value of $\lambda = 0.3$. For process 2, on the other hand, better results were obtained with a value of $\lambda = 0.1$, while for process 5 better results were obtained with a value of $\lambda = 0.5$. Notice that process 2 is the most complex workflow used in the experiment. For this reason, a lower smoothing constant works better for this model, since we need to consider a longer history to better predict the probability of the next performed action. Process 5, on the other hand was the most simple worflow, with few alternative paths and containing only a simple loop. The majority of variations in the log for this model corresponded to cycling in the loop in one of the alternative branches, so it was very easy to "compress" the sequences in the model and therefore only considering the last three observed actions was enough to predict the probability of the next performed action.

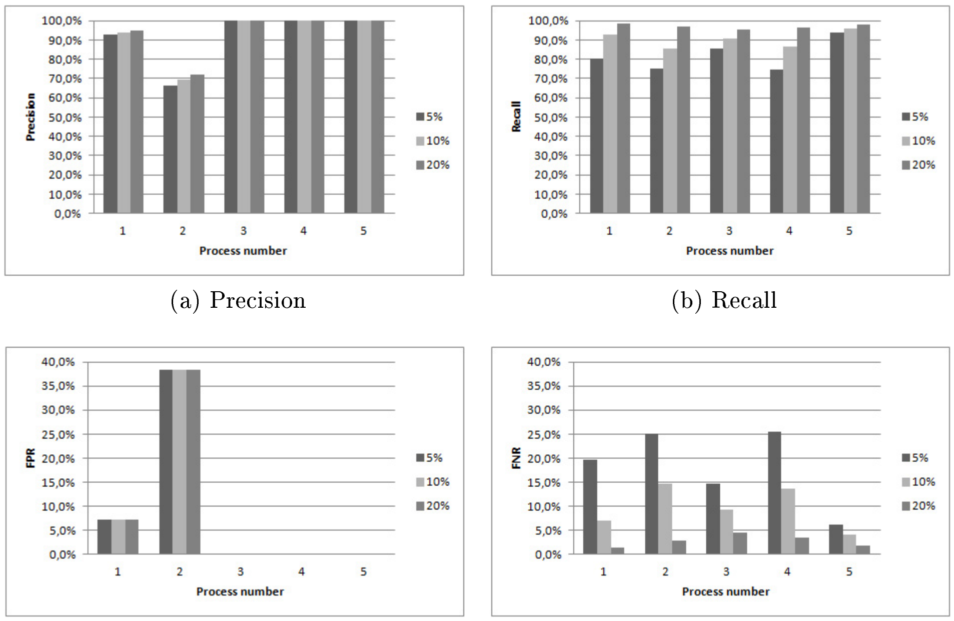
4.2 Recognition of abnormal behavior sequences

Next, we performed the classification of each sequence in each of the three logs (with 5%, 10% and 20% of noise). Figure 1 shows the results obtained for precision (a), recall (b), false positive rate (c), and false negative rate (d)

As expected, we can observe that the introduction of more noise in the sequences enables a better distinction of outliers from normal sequences. For processes 3, 4 and 5, we obtained optimal precision, that is, all the outliers detected were certainly abnormal sequences, and no normal sequences were deemed as outliers. For process 1, on the other hand, precision was over 90% for all noise levels, while for process 2, precision varied from 66.4% for the log containing sequences with 5% noise to 71.9% for the log containing sequences with 20% noise. In the case of process 2, 38.3% of the sequences recognized as abnormal corresponded in fact normal sequences.

Regarding recall, in the dataset with 5% of noise, our approach detected 80.2% of the outliers for process 1, 75% for process 2, 85.3% for process 3, 74.4% for process 4 and 93.8% for process 5. These values are increased to 98.5%, 97.2%, 95,4%, 96.6% and 98.2% respectively in the dataset containing 20% of noise.

The false negative rate refers to the percentage of outliers that were not detected by our approach. In the worst case, we failed to detect 25.6% of the abnormal sequences for process 4 in the dataset containing 5% of noise. When we increase the percentage of noise introduced to the sequences in the processes logs, we miss only 4.6% of abnormal sequences in the worst case.



(a) Precision (b) Recall

(c) False Positive Rate (d) False Negative Rate

Fig. 1: Outlier detection performance

5 Conclusions

In this article, we addressed the problem of automatically building a statistical model from different processes logs generated within the workflow of an organization. This model is then used to detect sequences of tasks that do not correspond to the expected behavior according to the workflow of the organization. This can be useful to detect employees that do not follow the normal workflow in order to assist them with the completion of the expected tasks.

We evaluated the proposed approach in a simulated scenario with five different processes workflows, when introducing different amount of noise in the sequences of the workflow log. We conclude that our approach is useful to detect abnormal behavior in the tasks performed, with an average precision of 91.8% when the executed sequences contain 5% of noise, 92.7% for 10% of noise, and 93.4% for 20% of noise. Regarding recall, our approach was able to detect on average 81.8% of the sequences containing abnormal behavior with 5% of noise, 90.3% of the sequences with 10% of noise, and 97.2% of the sequences with 20% of noise.

One limitation of our approach is that expressiveness of the model built is reduced since we are not reconstructing the underlying workflow from the logs. Nevertheless the graphical representation of the VOM model in the form of a PSA can give us an idea of contexts of tasks that determine the different transitions in the model. VOM models are also sensitive to the number of actions in the training sequences: if the provided training sequences are too short, we cannot take advantage of the variable order part of our models, since there is not enough statistical information to learn about. A similar problem occurs with the EMA computation. If sequences are too short, the memory of the model will not be activated and using a smoothing constant with higher values can lead to better prediction results.

References

1. van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Armentano, M., Amandi, A.: Modeling sequences of user actions for statistical goal recognition. User Modeling and User-Adapted Interaction In Press (2012)
3. Bouarfa, L., Dankelman, J.: Workflow mining and outlier detection from clinical activity logs. Journal of Biomedical Informatics 45(6), 1185–1190 (Dec 2012)
4. Burattin, A., Sperduti, A.: Plg: a framework for the generation of business process models and their execution logs. In: Proceedings of the 6th International Workshop on Business Process Intelligence (BPI 2010). Stevens Institute of Technology, Hoboken, New Jersey, USA (September 13 2010)
5. Chuang, Y.C., Hsu, P., Wang, M., Chen, S.C.: A frequency-based algorithm for workflow outlier mining. In: Kim, T.h., Lee, Y.h., Kang, B.H., Slezak, D. (eds.) Future Generation Information Technology, Lecture Notes in Computer Science, vol. 6485, pp. 191–207. Springer Berlin Heidelberg (2010)
6. Claes, J., Poels, G.: Merging Computer Log Files for Process Mining: an Artificial Immune System Technique, Lecture Notes in Business Information Processing, vol. 99, pp. 99–110. Springer (2012)
7. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Trans. Softw. Eng. Methodol. 7, 215–249 (July 1998)
8. Ghionna, L., Greco, G., Guzzo, A., Pontieri, L.: Outlier detection techniques for process mining applications. In: An, A., Matwin, S., Ras, Z., Slezak, D. (eds.) Foundations of Intelligent Systems, Lecture Notes in Computer Science, vol. 4994, pp. 150–159. Springer Berlin Heidelberg (2008)
9. Hunter, J.S.: The exponentially weighted moving average. Journal of Quality Technology 18(4), 203–209 (October 1986)
10. Kransdorff, A.: Corporate Amnesia: Keeping the Know-How in the Company. Butterworth Heinemann (1998)
11. Toon Jouck, B.D.: Generating artificial event logs to compare process discovery techniques. In: Proceedings of the 4th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2014). CEUR Workshop Proceedings, vol. 1293. Milan, Italy (November 2014)
12. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering workflow models from event-based data using little thumb. Integr. Comput.-Aided Eng. 10, 151–162 (April 2003)
13. Wen, L., Wang, J., Aalst, W.M., Huang, B., Sun, J.: A novel approach for process mining based on event types. J. Intell. Inf. Syst. 32, 163–190 (April 2009)