JMLR: Workshop and Conference Proceedings 11 (2010) 127–133 Workshop on Applications of Pattern Analysis

Modeling Knowledge Worker Activity

Tadej Štajner TADEJ.STAJNER@IJS.SI
Dunja Mladenić DUNJA.MLADENIC@IJS.SI
Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia

Editors: Tom Diethe, Nello Cristianini, John Shawe-Taylor

Abstract

This paper describes an approach to constructing a probabilistic process model representing knowledge worker activity out of a log of primitive events, such as e-mails, web page visits and document accesses. Firstly, we present the process of enriching the primitive events into abstract actions, executed in different contexts. We explain the process of obtaining both context and action for each event by clustering the events via two different views. Secondly, we present an application of probabilistic deterministic finite automata to model the transitions between consecutive actions within the same context and demonstrate the approach on real-world knowledge worker data for the purpose of understanding knowledge processes and demonstrating the feasibility of the proposed approach, where a process model is constructed out of low-level events.

1. Introduction

Imagine a scenario with a number of knowledge workers in an enterprise who are usually involved in several projects that require accessing different data sources, exchanging messages, browsing the Web etc. We then record activity on the level of complex events, such as, person P has accessed document D at time t. We will assume that each event is associated to a project and that it is possible to cluster the events so that we automatically identify which events belong to the same project. Each project has data collections associated to it - for instance, a set of people working on the project are related to a collection of documents they have written during that project. Identifying parts of data collections relevant for each project can then be approached as a clustering problem, where we are clustering the events enriched with features from documents and people.

In this framework we assume that knowledge workers while working on one project switch between different processes related to the project and that each process consists of a set of actions performed in a pattern. The ability to construct a process model allows for two classes of solutions: (1) assisting a knowledge worker by using context- and process-aware information delivery and (2) helping a manager or a process analyst in better understanding the process with the purpose of optimizing bottlenecks and decision support. This paper focuses on the latter âĂŞ enabling process analysts to have a capability to obtain a visual representation of the knowledge process (Gomez-Perez et al., 2009) out of plain event logs. The proposed approach is implemented in the TaskMiner system and applied to real-world data.

The rest of this paper gives problem formulation, description of the input data properties and details of the three subtasks of the proposed approach: context discovery, action discovery and process mining. The paper concludes with brief description of the related work, illustrative results demonstrating applicability of the proposed approach on real-world data and overall conclusions.

2. Problem formulation

The addressed problem can be formulated as follows: given a database, describing events in a business setting, such as sent e-mail messages and visited web sites, all executed in different contexts, produce a probabilistic temporal model that best describes the action patterns appearing in the event log. We obtain the model by solving three subtasks: context mining, action mining and process mining. Context mining and action mining can be solved independently, resulting with grouping of events that belong to the same context and the same action, respectively. Process mining then takes these as input, modeling sequences of actions that belong to the same context.

Context mining is the task where we want to discover different contexts that the knowledge worker is involved in. The contexts are obtained by performing clustering of events, where each cluster represents a distinct context in which the knowledge worker is working. In the addressed work environments, contexts most often correspond to projects or company clients, such as *Working on research project X* or *Proposal for client Y*.

Action Mining is the task where we wish to look at the events in a context-free manner and identify more general actions. It is performed by clustering of a context-free representation of events. When events are stripped of context-describing features and given additional meta-data, we are left with clusters, which describe generalized representations of events, such as *Send e-mail to group of co-workers*or *View intranet website*, which may occur in multiple contexts on different occasions.

Process mining is intended to show us the dynamics of the knowledge process of either a particular knowledge worker or an aggregate process model of an entire team. It gives us the probabilistic model of transitions between actions within a context.

3. Dataset

In this section, we describe data preprocessing - the process used to transform raw logs from different sources into a common TNT (text, network, time) event model (Grobelnik et al., 2009). We outline the required transformation steps and describe the additional background knowledge used. An actual example data set of real-world knowledge worker activities was collected from instrumenting the workstations of three knowledge workers within a large telecommunications company for two months, containing 15384 events. It contains several types of events: web page navigation events, opening or saving a document or reading, sending or replying to an e-mail message. We then generalize the different types of events

to a common framework of text, social network and, time. In the machine learning setting, we can look at these events as data points, containing the following features:

1. Content, associated with the event (i.e., text of a document or, a website, e-mail message)
2. Time and type of the event (i.e., navigate to a web page, view or send an e-mail)
3. The people, associated with the event (i.e., e-mail recipients, institutions)

The different transformations of the dataset for particular applications are described in the following sections. The actual clustering algorithm is k-means, used as follows: first, we represent data points as feature vectors including textual and social network data with an additional temporal dimension of the event. The similarity function is therefore defined as a weighted sum of text, social network and time similarities. More precise, individual similarity functions are:

Text: cosine similarity over TF-IDF feature vectors (each feature is a word);

Time: exponential decay of time difference - $sim_{time}(t_1, t_2) = 1 - \exp(c \cdot (t_2 - t_1))$, t_1 and t_2 being the timestamps of respecitve events and c being a damping coefficient.

Social network: cosine similarity over binary feature vectors (each feature is a person or some person's attribute, i.e. affiliation);

4. Context discovery

Context as defined in this paper is used as a term for grouping information for a particular need. To obtain such a grouping, we resort to automated ways of discovering contexts out of event logs. We can distinguish between different contexts by different content keywords, resources and people, involved in the events. Also, a knowledge worker is working in a single context for some period of time, producing multiple consecutive events. This means that the most appropriate features for context discovery are literal names and affiliations of people, contents of the document and the time of the event, since two events, appearing close in time are more likely from the same context.

5. Action discovery

We define actions as atomic steps in executing processes. The events that are logged are in fact manifestations of actions of a knowledge process being executed. This way, actions may be described as context-independent abstractions of events which would denote the event's intent. In practice, the algorithm to obtain actions is the same as the one to discover contexts with the difference in the feature sets. For instance, we display actions as feature patterns like *manager sent an email to a project partner* or *technical consultant prepared a proposal*. The following features are constructed to obtain an appropriate representation of events:

1. Person features using only their meta-data, such as organizational role (i.e., manager, researcher, administrative, domain expert), project role (i.e., project partner or not) or a descriptive role (i.e., academic, industry partner) âĂŞ without concrete identifiers.
2. Identify whether the event involved only one person, two people or a group.
3. Identify whether the people involved with the event are within the same institution, with another partner institution or from multiple different institutions.
4. Extract the named entities from the textual content and remove their mentions within the content. This is done to remove the references to concrete people or organizations which are more likely to be associated to a particular context, which we want to avoid.

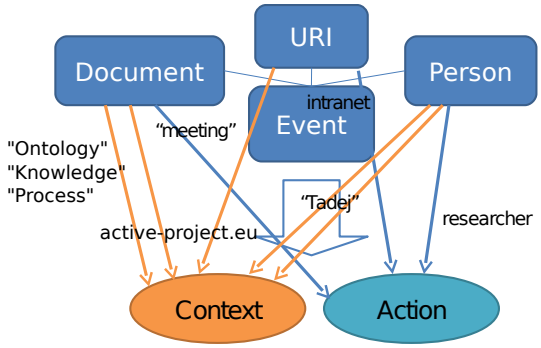

Figure 1: Different feature representations of the same data for context and action discovery

Figure 1 illustrates the two distinct data representations for clustering in order to obtain two different sets of clusters for the same event log.

6. Process mining

Process mining is a type of data mining which, by using specific techniques to mine large number of event logs, automatically (or semi-automatically) builds a model of the underlying process. The techniques and tools process mining provides are used in process management by enabling the discovery of process from event logs (Van Dongen et al., 2005) which can prove useful, on one hand, when no formal specification of the business process exists but knowing how it unfolds could be beneficial for optimizing it. On the other hand, when there is a formal definition of the process, it can be useful to control whether it is being followed at all. Our goal is to construct a simple as possbile process model without prior knowledge.

In the proposed approach, we use probabilistic deterministic finite automata as process models to model the transitions between consecutive actions within the same context. Given the same event log, let us assume that for each event, we already know its context and action. As a final step, we take all events in a given context and treat them as a sequence of transitions from one action to another. We then construct a probabilistic deterministic

finite automaton (Hingston, 2002) from those transitions. Since the obtained model can be very noisy and dense due to either irregularities in knowledge processes or due to noise in context or action discovery, we need to determine whether the existence of a particular transition between states is sensible enough so that the added complexity does not outweigh the improved coverage. A practical solution to this problem is to prune the model so that it is less complex but still retains useful coverage of the log while avoiding reporting of false discoveries. Given an error rate and a a sample size, we use statistical sequence mining techniques to determine constraints for inclusion of individual transitions in (Jacquemont et al., 2009). We apply the proportion constraint: Let $w = \langle x_1, \ldots x_l \rangle$ be a pattern of actions and q_0 the beginning action and $P(q, w)$ the probability that a path that starts in state q contains the pattern w. Given a risk factor α and a event log size N, the proportion constraint only allows the patterns which satisfy the following constraint: $P(P(q_0, w) > k)$ **iff** $k = z_{\alpha_1}\sqrt{\frac{P(q_0,w)-(1-P(q_0,w))}{N}}$, where z_{α_1} is the $(1-\alpha_1)$-percentile of the distribution of $p(w)$, a normal distribution in our case. A benefit of using a statistical approach is that the only parameter that the process analyst needs to specify is the risk factor, which corresponds to the expected false positive rate and is easier to understand than some arbitrary probability threshold.

To evaluate, we measure the predictability of the model in terms of the percentage of transitions that were valid within the obtained model, varying the number of actions (k in action mining) and the allowed error rate (α). The values were obtained as averages on five-fold cross-validation.

k / α	0.05	0.1	0.25	0.4	0.5
5	0.22	0.28	0.42	0.41	0.54
6	0.47	0.47	0.49	0.49	0.49
7	0.03	0.08	0.13	0.09	0.18
8	0.08	0.08	0.05	0.11	0.12
9	0.13	0.24	0.38	0.42	0.34
10	0.35	0.35	0.55	0.44	0.4
11	**0.58**	**0.6**	**0.58**	**0.6**	**0.59**
12	0.13	0.15	0.18	0.29	0.23
13	0.39	0.4	0.43	0.5	0.51
14	0.46	0.39	0.42	0.47	0.44
15	0.08	0.06	0.1	0.11	0.1

Table 1: The predictability of different process models with varying number of actions and pruning parameters.

To evaluate the quality of the obtained process models, we vary the k for action mining and α on model pruning. Table 1 shows that while the predictability varies quite a lot within different k values for action mining, it varies to a smaller degree across different allowed error rates. The latter behaviour is desirable; it suggests that pruning the model (and therefore simplifying it) does not have a too adverse effect on prediction performance.

As observed, the best performing model in this particular case with regard to k is when $k = 11$, exhibiting around predictability of roughly sixty per cent, as well as being relatively invariant to pruning.

The obtained models show some regularities. For instance, the transitions between actions related to internal communication inside the company have higher transition probabilities between each other than to other actions. Looking deeper, we are able to observe is that the contexts themselves are often segmented into communication sessions and content consumption and authoring sessions. This manifests itself as a higher transition probability from one communication-related action, such as *View email from consortium partner* to another, such as *Reply to consortium partner*. On the other hand, content consumption and authoring actions, such as web browsing, reading documents tend to be longer, less frequently interrupted by communication events. Using this approach, one is able to see whether a knowledge worker, while still working in a single context, is interrupted by communication events too often or is handling too many contexts. Comparing models for different knowledge workers can also demonstrate whether different people employ different work patterns, whereas aggregating several models from several knowledge workers emphasizes the shared patterns.

Measurements show that selecting different k values has a very big effect on the predictability of the model. In terms of interpretation of the model, it is often worth pruning the models as it does not have a too adverse effect on predictability.

7. Related work

The relationship between the process and the context has also been discussed in more specific domains, such as medicine (Ghattas et al., 2010), where the authors focus on identifying contexts, given that they already have knowledge of a formal process, where our approach focuses on discovering the models out of event logs without an pre-existing process model. Another interesting approach to identifying different contexts using the sequence information has also proven to be useful especially when dealing with multiple simultaneous processes being executed (Bose and van der Aalst, 2009).

Process mining in itself is often discussed as a stand-alone topic of business process mining (Van Dongen et al., 2005), where the process analyst has access to higher level events within a business information system, meaning that the contexts and actions of the events are already given. Business processes are often modeled via more expressive Petri nets, whereas we have found finite automata (Hingston, 2002) to be more suited to the low-level events in the data. This sort of approach was also successfully used in modeling traffic flows within a city (Jacquemont et al., 2009).

8. Conclusions and future work

This paper described the three-phase approach for constructing probabilistic process models from raw event logs obtained from instrumenting common knowledge worker tools. We have found that the output of action mining has a big influence on the predictability of the obtained process model. In the most optimal scenario, we observe relatively high predictability of around sixty per cent. Moreover, we show that pruning of probabilistic process

models does not have a too adverse effect on predictability and proves to be an effective way of making a compromise on interpretability vs. predictability.

Future research in this area will include using a complex graph representation of data so that we can avoid flattening the relational data into features. To take advantage of the structural information and to correctly handle differences in distributions of features across people, events and resources, present in the multi-relational representation, we will employ multi-relational clustering algorithms which are able to handle such datasets. This sort of approach may not only improve the clustering quality, but also report the clusterings of people and resources, respectively.

Application-wise, activity will focus on two parts: the first one is introducing a new user-study group of the telecommunications company which will try to take advantage of the contextual information delivery mechanisms, enabled by context mining, coupled with automated context detection. The second goal is expanding the applicability of action and process mining into more than just providing an analytic environment for managers, but also to use the obtained models to aid user with predicting the next action or suggesting action-specific resources or text fragments when editing a document or a message.

Acknowledgments

This work was supported by the IST Programme of the EC under ACTIVE (IST-2008-215040) and PASCAL2 (IST-NoE-216886).

References

R.P.J.C. Bose and W.M.P. van der Aalst. Context aware trace clustering: Towards improving process mining results. In *SIAM International Conference on Data Mining*, pages 401–412, 2009.

J. Ghattas, M. Peleg, P. Soffer, and Y. Denekamp. Learning the context of a clinical process. In *Business Process Management Workshops*, pages 545–556. Springer, 2010.

J.M. Gomez-Perez, M. Grobelnik, C. Ruiz, M. Tilly, and P. Warren. Using task context to achieve effective information delivery. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, pages 1–6. ACM, 2009.

M. Grobelnik, D. Mladenić, and J. Ferlež. Probabilistic Temporal Process Model for Knowledge Processes: Handling a Stream of Linked Text. *Proceedings of SiKDD 2009 (Conference on Data Mining and Data Warehouses)*, 2009.

P. Hingston. Using finite state automata for sequence mining. *Australian Computer Science Communications*, 24(1):105–110, 2002.

S. Jacquemont, F. Jacquenet, and M. Sebban. Mining probabilistic automata: a statistical view of sequential pattern mining. *Machine Learning*, 75(1):91–127, 2009.

BF Van Dongen, AKA De Medeiros, HMW Verbeek, A. Weijters, and WMP Van der Aalst. The ProM framework: A new era in process mining tool support. *Applications and Theory of Petri Nets 2005*, pages 444–454, 2005.