Discovering Process Models: a Multi-Relational Approach

Antonio Turi, Annalisa Appice, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
{turi,appice,ceci,malerba}@di.uniba.it

Abstract. The automatic discovery of process models can help to gain insight into various perspectives (e.g., control flow or data perspective) of the process executions traced in an event log. Association rule mining offers a means of building a human understandable representation of these process models. The variety of activities and actors involved in a process execution demands for a relational (or structural) representation of process execution traces. This paper describes the application of a distributed multi-relational method for association rule discovery to process mining. The viability of the proposed approach is assessed on a huge log of process executions collected by a private company in the space of three years. In particular, the method discovers frequent associations among activities and/or actors of a process execution.

1 Introduction

Qualitative patterns such as human interpretable association rules are particularly interesting for business applications which aim to discover relationships between events (activities and their actors) and describe the system behavior of several executions (cases) of a candidate business process.

Association rule discovery poses several difficulties when they are discovered from data which describe process executions registered in an event log. First of all, a log collects objects which belong to different data types (cases, activities and actors). Separate data types are modeled as several relational data tables (one for each data type). This leads to distinguish between the reference objects of analysis (cases) and other task-relevant objects (activities and actors), and to represent their interactions. Classical algorithms for association rule discovery [1] assume that data are stored in a single table of a relational database, hence they do not make any distinction between reference objects and task-relevant objects, nor they allow the representation of any kind of interaction. Second, events stored in a log are marked from a timestamp. This timestamp indicates the time of occurrence and implicitly defines a total temporal order over events. Temporal autocorrelation requires that the effect of a property at any event may not be limited to the specific event. Third, some domain knowledge (e.g. the definition of the ordering relation between events) may be available, which can be profitably exploited only if some inferential mechanisms typical of a theorem prover are used by the data mining algorithms.

These challenges are naturally faced by resorting to methods developed in multi-relational data mining [7], where data are assumed to be spanned in several data tables (or relations) of a relational database and relational patterns (i.e. patterns which may involve several relations at once) are discovered. SPADA [12] is a multi-relational data mining method that discovers relational association rules. In the case of log data, rules discovered by SPADA capture interesting associations between process executions (reference objects) and events (task-relevant objects). Task-relevant objects are represented at several levels of a generalization hierarchy (e.g. an actor can be an administrator, a viewer or a user), hence rules are discovered at several levels of granularity. In addition, SPADA can use some available domain knowledge during the discovery process. The main limitation of SPADA is the high computational complexity which makes the analysis of large databases practically unfeasible. To overcome this limitation, a distributed version of SPADA has been realized, which produces "approximate" association rules by sampling the original data, by running SPADA on computational nodes of a GRID to analyze each sample independently, and finally by collecting local results of each SPADA execution in order to find approximate global results [2].

In order to prove the viability of the proposed multi-relational approach also on real cases, in this paper we briefly describe the distributed multi-relational method for association rule discovery (Section 2) and then we present an application to mining a very large data set derived from a real event log (Section 3). Finally, some conclusions are drawn.

2 Process Models in Multi-Relational Data Mining

Studies for association rule discovery in Multi-Relational Data Mining [12] are rooted in the field of Inductive Logic Programming (ILP) [14]. In ILP both relational data and relational patterns are expressed in a first-order logic and the logical notions of generality order and of the downward/upward refinement operator on the space of patterns are used to define both the search space and the search strategy. In the specific case of SPADA, properties of both reference and task relevant objects are represented as the extensional part D_E of a deductive database D [6], while the domain knowledge is represented as a normal logic program which define the intensional part D_I of the deductive database D.

In the application of SPADA to process mining, the extensional database stores information on the event log (e.g., process executions and actors) while the intensional database includes the definition of the relation *before* which makes explicit the execution order of activities that is implicit in the timestamp of each event. An example of possible definition of the relation *before* is the following:

$$\begin{aligned}&before(A1, A2) \leftarrow event(C, A1), event(C, A2), A1 \neq A2,\\&\quad time(A1,T1), time(A2,T2), T1<T2,\\&\quad not(activity(C, A), A\neq A1, A\neq A2, time(A,T), T1<T, T<T2)\end{aligned}$$

which defines the direct successor relation between two events.

The set of ground atoms in D_E is partitioned into a number of non-intersecting subsets $D[e]$ each of which includes facts concerning the activities and actors involved in a specific process execution e. This means that SPADA discovers frequent patterns across the different process executions, which are the *units of analysis*. This partitioning of D_E is coherent with the individual-centered representation of training data [4], which has both theoretical (PAC-learnability) and computational advantages (smaller hypothesis space and more efficient search). Fragments of the process models underlying the generation of the various logged executions are expressed in the form of relational association rules:

$$process(P) \Rightarrow \mu(P) \quad [s, c],$$

where $process(P)$ is the atom that identifies a process execution, while $\mu(P)$ is a conjunction of atoms which provide a description of a fragment of the process execution P. Each atom in $\mu(P)$ describes either the before relation between activities or the participation of an actor to an activity or a property of an actor, activity and/or process execution. An example of discovered association rule is the following:

P1: *process(P) ⇒*
activity(P,A), is_a(A,activity), before(A,B), is_a(B,activity),
actor(A,C), is_e(C, actor). [s=63%, c=100%]

The support s estimates the probability $p(process(P) \bigcup \mu(P))$ on D. This means that s% of the units of analysis $D[e]$ are covered by $process(P) \bigcup \mu(P)$, that is a substitution $\theta = \{P \leftarrow e\} \cdot \theta_1$ exists such that $process(P) \bigcup \mu(P)\theta \subseteq D[e]$. The confidence c estimates the probability $p(\mu(P)|process(P))$.

SPADA performs both an intra-level search and an inter-level search. The former is based on the classical levelwise method described in [13], with the variant that the syntactic ordering between patterns is based on θ-subsumption [15]. In the intra-level search all the task-relevant objects in a pattern belong to the same level l of the generalization hierarchies. In the inter-level search, SPADA takes advantage of statistics computed at a level l when it searches in the space of more specific task-relevant objects at level $l+1$. By descending through a hierarchy it is possible to view the same activity or actor at different levels of abstraction (or granularity) and discover multi-level association rules. For example, the followingassociation rule:

P2: *process(P) ⇒*
activity(P,A), is_a(A,namemaker), before(A,B), is_a(B,workflow),
actor(A,C), is_e(C, admin). [s=55%, c=100%]

provides us with better insight into the nature of activities and actors than $P1$.

The application of SPADA to process mining is not straightforward, since the high computational cost of the search and the usage of an in-memory deductive database do not allow for mining massive event logs. *sGSPADA* [2] overcomes computational limits of SPADA by distributing and (possibly) parallelizing the discovery of local process models on random multi-samples of the original set of units of analysis and then by deriving an approximation of the set of "exact" global association rules (i.e. association rules to be discovered on the entire database) from the various sets of local rules. "Approximate" global association rules are those which are discovered in at least k of the n sample databases. Support and confidence are estimated by averaging the values of support and confidence of local association rules.

Random multi-sampling generates n samples with replacement, each of which includes a percentage p of the data in the original dataset. The samples generated are not a partition, so even 10 samples with $p = 10\%$ do not generally correspond to the entire dataset. This sampling procedure is similar to that used in bootstrap estimation of a parameter (e.g., predictive accuracy of a classifier) [8], as well as in some ensemble data mining methods, such as bagging [5], which combine multiple models to achieve better prediction accuracy than any of the individual models could on their own.

As observed by Zaki et al. [17] sampling can speed up the mining process by more than an order of magnitude by reducing I/O costs and drastically shrinking the number of transactions to be considered. Moreover, when training data are kept in main memory as in SPADA, sampling is the only way to make their analysis workable. By working on sampled databases, it is also possible to distribute the computation on a Grid and to parallelize the execution the data mining algorithm. The problem of discovering spurious local patterns can be mitigated by increasing the number of samples, so that the set of "approximate" global patterns derived from local patterns is more likely representative of the set of "exact" global patterns.

3 Mining Process Models from an Event Log

Experiments are performed on a real event log provided by THINK3 Inc[1], one of the global player in the market of solutions for the Product Lifecycle Management, whose mission is to help manufacturers optimizing their entire product development processes and to help industrial designers creating better, more innovative products.

3.1 Data Description

THINK3 data traces 353,490 process executions of one of its customers. The period under analysis is from April 7th, 2005 to January 10th, 2007 for a total of 1,035,119 activities and 103 promoters. Taxonomic knowledge on activities and actors is encoded in two distinct hierarchies reported in Figure 1. For example, "muller" is an "administrator", "altendorfer" is a "user", "administrator" is an "originator" and so on. Both a user and an administrator are generic actors.

[1] http:// www.think3.com/en/default.aspx

```
activity [1035119]
+ − − administrator tools [131]
|          + −− o665536, o713252,...
+ − − workflow [919052]
|          + −− o0,o1,o2,...
+ − − namemaker [106839]
|          + −− o9,o12,...
+ − − delete [2767]
|          + −− o318,o413,...
+ − − deleteEnt [2354]
|          + −− o388491,o391204,...
+ − − prpDelete [471]
|          + −− o339710,o340636,...
+ − − prpSmartDelete [53]
|          + −− o794996,o794997,...
+ − − prpModify [34]
|          + −− o488390,o522414,...
+ − − cast [1430]
           + −− o292,o491,...

actor [108]
+ − − user [103]
|          + −− altendorfer,amaeder,andrea,...
+ − − administrator [3]
|          + −− mueller,cma,admin
+ − − viewer [2]
|          + −− fenzl, wimmer
```

Fig. 1. Hierarchies on activities and actors involved in the process executions traced in the THINK3 event log.

For each activity a textual description is registered in the event log, while for each actor a working group is defined. In this experiment, we have thirteen distinct descriptions of the activities and thirty-three distinct groups of promoters. The extensional database D_E includes 4,374,840 ground facts, while the intensional part D_I includes the definition of the predicate *before*, which takes into account temporal autocorrelation of events. Additional predicates are intensionally defined to group similar activities. For example, the following clauses:

release(X) $\leftarrow$ *description(X,freigabe).*
release(X) $\leftarrow$ *description(X,freigabe_h).*
release(X) $\leftarrow$ *description(X,freigabe_j).*
release(X) $\leftarrow$ *description(X,freigabe_m).*

define the predicate "release" which describe the activity of release ("freigabe" in German) independently from the release type (H, J or M). Similarly, other clauses in D_I define the predicates *pruefung*, *techaend*, *cancelled*, *construction*, *ktgprocess*, *musterbau*, *nullserie*, *techniche*, *tiffprocess*, *undermodify* and *workinprogress* which describe an activity.

3.2 Process Model Discovery

sGSPADA is run by randomly extracting n=100 sample databases with p=1%. The discovery of the local multi-level frequent relational patterns is parallelized on 100 nodes. The size of the original dataset prevents SPADA from processing units of analysis altogether.

Different minimum support thresholds are defined for each level, such that the higher the level (i.e., the more abstract the task-relevant objects involved in the pattern), the higher the support (i.e., the more selective is the discovery process). In particular we set the following SPADA's parameters: $minsup[1] = 0.25$, $minsup[2] = 0.1$, $minsup[3] = 0.01$ and $max\_len\_path = 14$. The last parameter defines the maximum number of atoms in a frequent pattern considered by SPADA during its search. With the thresholds defined above, there are no frequent patterns with more than twelve atoms, this means that in this case study, sGSPADA returns the set of all approximate global association rules.

Approximate global association rules are reconstructed from the local ones by varying k from 1 to 100. Their number is reported in Table 1 and, as expected, it decreases when k increases. The average number of local frequent patterns (at any level) discovered on each sample is 673.11, while the standard deviation is relatively small (53.47). As reported in the last column of Table 1, 369 local frequent patterns (about 54% on average) are common to all samples.

Table 1. Number of approximate global frequent rules found by varying k in [1,100].

k	1	10	20	30	40	50	60	70	80	90	100
#P	**1244**	**1043**	**820**	**747**	**669**	**619**	**574**	**539**	**498**	**468**	**369**

An example of process model discovered at level $l = 2$ is the following:

P3: *process(P)* ⇒ *activity(P,A), is_a(A,workflow), before(A,B), B≠A, is_a(B,workflow), before(B,C), C≠B, C≠B, is_a(C,workflow), actor(A,D), is_a(D,user), workinprogress(A), release(C), construction(B).*

P3 covers 19,789 process executions and is found for $k = 90$. The global support computed by averaging over the support of all local frequent rules equal to P3 is 20.78% (obviously, it is greater than $minsup[2]$). This rule reports the execution order between three activities (A, B and C) within a process instance (P). Both A, B and C are workflow activities, but A is described as a *work in progress*, B as *release* and C as *construction*. The actor of A is a simple user (D).

By descending the hierarchy of actors, sGSPADA discovers the following rule:

P4: *process(P)* ⇒ *activity(P,A), is_a(A,workflow), before(A,B), B≠A, is_a(B,workflow), before(B,C), C≠B, C≠B, is_a(C,workflow), actor(A,D), is_a(D,andrea), workinprogress(A), release(C), construction(B).*

which provides us with a deeper insight on the actor of the work-in-progress, who is identified as *andrea*. This rule, however, covers only 1,139 process executions and has a global support of 1.26.

4 Related Works and Conclusions

Process mining targets the automatic discovery of knowledge from massive event logs. In this work, we investigate the discovery of association rules as a means to extract a human interpretable representation of process models. Association rules are intended as a means to capture the typical order execution between activities (control perspective) and, at the same time, they model the possible associations among the properties of the process, activities and actors (data perspective). In this work, we considered a multi-relational approach to association rule discovery in order to take into account the relational structure of data, since several activities and/or actors may be involved in the same process execution.

Recently, the multi-relational or ILP approaches to building business process models from event logs is receiving attention. Goedertier et al. [10] have faced the task of predicting whether, given the state of a process instance, a particular state transition can occur, by learning relational classification rules. The representation formalism considered in this work is Event Calculus, a first-order logic that elegantly captures the time-varying nature of facts. Learning is based on both positive information (possible transitions) and negative information (prohibited transitions). When no negative information is actually available in the logs, it is artificially generated by means of a sort of closed-world assumption (no pair of similar traces exists such that the transition of interest occurs). Similarly, Lamma et al. [11] have considered both compliant (positive information) and non compliant (negative information) execution traces and adapt the algorithm ICL [16] to learn constraints among activities (integrity constraints) expressed as logical formulas. In practice, the main problems of both methods are the reliable provision of negative information and their scalability to huge event logs.

Temporal data mining methods can also be applied to the discovery of the execution order of activities. Giannotti et al. [9] have proposed a paradigm to extract sequential patterns where each event transition has a temporal annotation. This paradigm has been applied to process logs [3] in order to discover the execution order of activities or actors. The analogy with the problem solved by sGSPADA stops here, since the proposed framework cannot capture the possible associations between the properties of the actors and the activities involved in a process, nor does it take into account hierarchies on activities and actors.

Future work on sGSPADA will investigate the integration of discovered fragments of process models in order to build more complex models, where parallel execution of processes is also allowed. We also plan to develop a plug-in for the extendible environment ProM 5.0[2].

[2] http://prom.sf.net/

Acknowledgments

This work is in partial fulfillment of the research goals of the projects DIPIS (DIstributed Production as Innovative System) and TOCAI.it (Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in interneT). The authors wish to thank THINK3 Inc for having provided process data used in the experiments.

References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *International Conference on Management of Data*, pages 207–216, 1993.
2. A. Appice, M. Ceci, A. Turi, and D. Malerba. Sampling very large databases for parallel and distributed relational frequent pattern discovery. In *First International Workshop on Ubiquitous Knowledge Discovery Workshop*, 2008.
3. M. Berlingerio, F. Giannotti, M. Nanni, and F. Pinelli. Temporal analysis of process logs: a case study. In S. Gaglio, I. Infantino, and D. Saccà, editors, *Proceedings of the Sixteenth Italian Symposium on Advanced Database Systems, SEBD 2008*, pages 430–437, 2008.
4. H. Blockeel and M. Sebag. Scalability and efficiency in multi-relational data mining. *SIGKDD Explorations*, 5(1):17–30, 2003.
5. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
6. S. Ceri, G. Gottlob, and L. Tanca. *Logic Programming and Databases.* Springer-Verlag New York, Inc., New York, NY, USA, 1990.
7. S. Džeroski and N. Lavrač. *Relational Data Mining.* Springer-Verlag, 2001.
8. B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, London, 1994.
9. F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining sequences with temporal annotations. In H. Haddad, editor, *Proceedings of the 2006 ACM Symposium on Applied Computing, SAC 2006*, pages 593–597. ACM, 2006.
10. S. Goedertier, D. Martens, B. Baesens, R. Haesen, and J. Vanthienen. A new approach for discovering business process models from event logs. In *Technical Report.* KBI 0716, 2007.
11. E. Lamma, P. Mello, F. Riguzzi, and S. Storari. Applying inductive logic programming to process mining. In H. Blockeel, J. Ramon, J. W. Shavlik, and P. Tadepalli, editors, *International Conference on Inductive Logic Programming,ILP 2007*, volume 4894 of *LNCS*, pages 132–146. Springer-Verlag, 2007.
12. F. A. Lisi and D. Malerba. Inducing multi-level association rules from multiple relations. *Machine Learning*, 55(2):175–210, 2004.
13. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
14. S. Muggleton. *Inductive Logic Programming.* Academic Press, London, 1992.
15. G. Plotkin. A note on inductive generalization. volume 5 of *Machine Intelligence*, pages 153–163. Edinburgh University Press, Edinburgh, 1970.
16. L. D. Raedt and W. V. Laer. Inductive constraint logic. In K. P. Jantke, T. Shinohara, and T. Zeugmann, editors, *International Conference on Algorithmic Learning Theory, ALT 1995*, volume 997 of *LNCS*, pages 80–94. Springer, 1995.
17. M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Seventh International Workshop on Research Issues in Data Engineering, RIDE 1997*, 1997.