Mining Probabilistic Declarative Process Models

Elena Bellodi and **Fabrizio Riguzzi** and **Evelina Lamma**
ENDIF – Università di Ferrara – Via Saragat, 1 – 44100 Ferrara, Italy.
{elena.bellodi,evelina.lamma,fabrizio.riguzzi}@unife.it

Organizations usually rely on a number of processes to achieve their mission, which describe the way resources are exploited. Formal ways of representing business processes have been studied in the so-called area of Business Processes Management (BPM). Recently, the problem of automatically mining a structured description of a business process directly from real data has been studied by many authors. The data consist of execution traces of the process and are collected by information systems which log the activities performed by the users. This problem has been called Process Mining. Recently new declarative languages have been proposed to express only constraints on process execution.

In particular, $\mathcal{S}$CIFF adopts first-order logic in order to represent the constraints. A trace t is a sequence of events, described by a number of attributes. A bag of process traces L is called a *log*. The aim of Process Mining is to infer a process model from a log. A process trace can be represented as a logical interpretation (set of ground atoms): each event is modeled with an atom whose predicate is the event type and whose arguments store the attributes. A process model in $\mathcal{S}$CIFF language is a set of Integrity Constraints (ICs). An IC, C, is a logical formula of the form

$$\begin{aligned} Body \rightarrow \exists(ConjP_1) \vee \ldots \vee \exists(ConjP_n) \\ \vee \forall\neg(ConjN_1) \vee \ldots \vee \forall\neg(ConjN_m) \end{aligned} \tag{1}$$

where $Body$, $ConjP_i$ $i = 1, \ldots, n$ and $ConjN_j$ $j = 1, \ldots, m$ are conjunctions of literals built over event atoms, over predicates defined in the background or over built-in predicates. An IC C is true in an interpretation if, for every substitution θ for which $Body$ is true in it, there exists a disjunct $\exists(ConjP_i)$ or $\forall\neg(ConjN_j)$ that is true in it. In this case the trace t is *compliant* with C. The theory (or model) composed of all the ICs must be such that all the ICs are true when considering a positive trace and at least one IC is false when considering a negative one. The algorithm Declarative Process Model Learner (DPML) finds an IC theory solving the learning problem.

At this point we investigate the possibility of encoding probabilistic information in the IC theory with Markov Logic, a language extending first-order logic. ML allows to attach weights to ICs by means of the Alchemy system. In the infinite-weight limit, ML reduces to standard first-order logic. The resulting set of couples (weight,formula) is called Markov Logic Network (MLN). A set of ICs can be seen as a "hard" theory: if a world violates even one formula, it is considered impossible; in ML it is less probable, but not impossible. The weight associated to each formula reflects how strong the constraint is. The IC (1) corresponds to the following ML formula:

$$\begin{aligned} Body \wedge \neg(ConjP_1) \wedge \ldots \wedge \neg(ConjP_n) \\ \wedge(ConjN_1) \wedge \ldots \wedge (ConjN_m) \rightarrow Neg(i) \end{aligned} \tag{2}$$

where $Neg(i)$ means that trace i is negative.

Our goal is to demonstrate that the combined use of DPML, for learning an IC theory, and Alchemy, for learning weights for formulas, produces better results than the sharp classification realized by the $\mathcal{S}$CIFF theory. The experiments based on a real dataset of university students careers, where positive traces are students who graduated, and negative ones are students who did not finish their studies. First we induced ten SCIFF theories using a ten-fold cross-validation. Then we assigned weights to the learned theories translated into ML, with an Alchemy algorithm. Ten MLNs were also generated from the learned theories by assigning the very large weight $1e + 10$ to all the clauses, in order to approximate a purely logical theory. Then we computed the probability of each test trace of being negative, i.e. probability of the atoms $Neg(I)$ with I representing students' id in the test dataset, by running the belief propagation inference algorithm both on the MLNs with learned weights and on the MLNs with pseudo--infinite weights. Finally, the average area under the ROC curve (AUC) was computed: it is a measure for evaluating the classification performances of algorithms with respect to accuracy. The sharp MLN achieved an average AUC of 0.7107528, while the weighted MLN 0.7227286. We also applied a one-tailed paired t test: the null hypothesis that the two algorithms are equivalent can be rejected with a probability of 90.58%.

Conclusions. The probabilistic classification made with Alchemy is more accurate than the pure logical one made only by $\mathcal{S}$CIFF .