An AI Planning System for Data Cleaning

Roberto Boselli[1,2], Mirko Cesarini[1,2], Fabio Mercorio[1,2], and Mario Mezzanzanica[1,2]

[1] Dept. of Statistics and Quantitative Methods, Univ. of Milano-Bicocca, Italy
[2] CRISP Research Centre, Univ. of Milano-Bicocca, Italy

Abstract. Data Cleaning represents a crucial and error prone activity in KDD that might have unpredictable effects on data analytics, affecting the believability of the whole KDD process. In this paper we describe how a bridge between AI Planning and Data Quality communities has been made, by expressing both the data quality and cleaning tasks in terms of AI planning. We also report a real-life application of our approach.

Keywords: AI Planning, Data Quality, Data Cleaning, ETL

1 Introdution and Motivation

A challenging issue in data quality is to automatically check the quality of a source dataset and then to identify cleaning activities, namely a sequence of actions able to cleanse a dirty dataset. Data quality is a domain-dependent concept, usually defined as "fitness for use", thus reaching a satisfying level of data quality strongly depends on the analysis purposes. Focusing on *consistency*, which can be seen as "the violation of semantic rules defined over a set of data items" [1], the state-of-the-art solutions mainly rely on functional dependencies (FDs) and their variants, that are powerful in specifying integrity constraints. Consistency requirements are usually defined on either a single tuple, two tuples, or a set of tuples [4]. Though the first two kind of constraints can be modelled through FDs, the latter one requires reasoning with a (finite but variable in length) set of data items (e.g., time-related data), and this makes the use of FD-based approaches uneffective (see, e.g., [10,4]). This is the case of *logitudinal data* (aka *historical or time-series data*), which provide knowledge about a given subject, object or phenomena observed at multiple sampled time points. In addition, it is well known that FDs are expressive enough to model static constraints, which evaluate the current state of the database, but they do not take into account how the database state has evolved over time [3]. Furthermore, though FDs enable the detection of errors, they cannot be used as guidance to fix them [9].

In such a context graphs or tree formalisms are deemed also appropriate to model the *expected* data behaviour, that formalises how the data should evolve over time for being considered as consistent, and this makes the exploration-based technique (as AI Planning) a good candidate for the data quality task. The idea that underlies our work is to cast the problem of checking the consistency of a set of data items as a planning problem. This, in turn, allows using

off-the-shelf AI planning tools to perform two separated tasks: (i) to catch inconsistencies and (ii) to synthesise a sequence of actions able to cleanse any (modelled) inconsistencies found in the data. In this paper we summarise results from our recent works on data consistency checking [15] and cleaning [2,14].
AI Planning at a Glance. Planning in Artificial Intelligence is about the decision making performed by computer programs when trying to achieve some goal. It requires to synthesise a sequence of actions that will transform a system configuration, step by step, into the desired one (i.e., the goal state). Roughly, planning requires two main elements: (i) *the domain*, i.e., a set of states of the environment S together with the set of actions A specifying the transitions between these states; (ii) *the problem*, that consists of the set of facts whose composition determine an initial state $s_0 \in S$ of the environment, and a set of facts $G \subseteq S$ that models the goals of the planning taks. A solution (aka *plan)* is a bounded sequence of actions $a_1, \ldots, a_n$ that can ben applied to reach a goal configuration. Planning formalisms are expressive enough to model complex temporal constraints, then a cleaning approach based on AI planning might allow domain experts to concentrate on *what* quality constraints have to be modelled rather than on *how* to check them. Recently, AI Planning contributed to the trace alignment problem in the context of Business Process Modelling [5].

2 A Data Cleaning Approach framed within KDD

Our approach requires to map a sequence of events as actions of a planning domain, so that AI planning algorithms can be exploited to find inconsistencies and to fix them. Intuitively, let us consider an events sequence $\epsilon = e_0, e_2, \ldots, e_{n-1}$. Each event e_i will contain a number of observation variables whose evaluation determines a snapshot of the subject's *state*[1] at time point i, namely s_i. Then, the evaluation of any further event e_{i+1} might change the value of one or more state variables of s_i, generating a new state s_{i+1}.

We encode the expected subjects' behaviour (the so-called *consistency model*) as a transition system. A consistent trajectory represents a sequence of events that does not violate any consistency constraints. Given a ϵ event sequence as input, the planner deterministically determines a trajectory $\pi = s_0 e_0 s_1 \ldots s_{n-1} e_{n-1} s_n$ on the finite state system explored (i.e., a *plan*) where each state s_{i+1} results by applying event e_i on s_i. Once a model describing the evolution of an event sequence has been defined, we detect quality issues by solving a planning problem where a consistency violation is the goal condition. If a plan is found by a planning system, the event sequence is marked as inconsistent in the original data quality problem. Our system works in three steps (Fig. 1).

Step 1 [Universal Checker] We simulate the execution of *all* the event sequences - within a finite-horizon - summarising all the inconsistencies found during the exploration[2] into an object, we call *Universal Checker* (UCK), that represents a taxonomy of the inconsistencies that may affect a data source. The

[1] A value assignment to a set of finite-domain state variables

[2] Notice that this task can be accomplished by forcing the planner to continue the search even if a goal has been found.

UCK computed can be seen as a list of tuples (id, s_i, a_i), that specifies the inconsistecy with id might arise in a state s_i as consequence of applying a_i.

Step 2. [Universal Cleanser] For any given tuple (id, s_i, a_i) of the Universal Checker, we construct a new planning problem which differs from the previous one in terms of both initial and goal states: (i) the new initial state is s_i, that is a consistent state where the event e_i can be applied leading to an inconsistent state s_{i+1}; (ii) the new goal is to be able to "execute action a_i". Intuitively, a *cleaning action sequence* applied to state s_i transforms it into a state s_j where action a_i can be applied without violating any consistency rules. To this end, the planner *explores* the state space and *collects* all the optimal corrections according to a given criterion. The output of this phase is a *Universal Cleanser*. Informally, it can be seen as a set of policies, computed off-line, able to bring the system to the goal from any state reachable from the initial ones (see, e.g., [8,12]). In our context, the universal cleanser is a lookup table that returns a sequence of actions able to fix an event e_i occurring in a state s_j.

Step 3 [Cleanse the Data] Given a set of event sequences $D = \{\epsilon_1, \ldots, \epsilon_n\}$ the system uses the planner to verify the consistency of each ϵ_i. If an inconsistency is found, the system retrieves its identifier from the Universal Checker, and then selects the cleaning actions sequence through a look-up on the Universal Cleanser.

The Universal Cleanser presents two important features that makes it effective in dealing with real data: first, it is synthesised off-line and only summarises cost-optimal action sequences. Clearly, the cost function is domain-dependent and usually driven by the purposes of the analysis (we discussed how to select different cleaning alternatives in [14,13]). Second, the UC is *data-independent* as it has been synthesised by considering *all* the (bounded) event sequences, thus *any* data sources confom to the model can be handled. Our approach has been implemented on top of the UPMurphi planner [6,7].

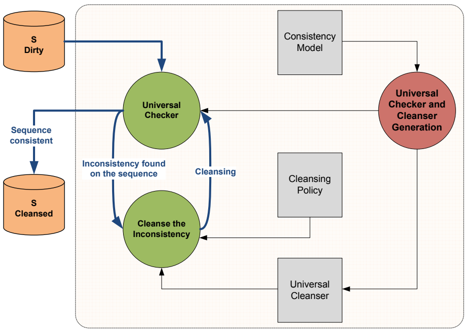

Fig. 1: A graphical representation of the Consistency Verification and Cleaning Process.

Real-life Application[3]. Our approach has been applied to the *mandatory communication*[4] domain, that models labour market data of Italian citizens at regional level. Here, inconsistencies represent career transitions not permitted by the Italian Labour Law. Thanks to our approach, we synthesised both the Universal Checker and Cleanser for the domain (i.e., 342 distinct inconsistencies found and up to 3 cleaning action sequence synthesised for each). The system has

[3] This work was partially supported within a Research Project granted by the CRISP Research Centre and Arifl Agency (Regional Agency for Education and Labour

[4] The Italian Ministry of Labour and Welfare: Annual report about the CO system, available at http://goo.gl/XdALYd last accessed may 2017

been employed within the KDD process that analysed the real career sequences of 214,432 citizens composed of 1,248,751 mandatory notifications. For details about the quality assessment see [15] whilst for cleaning details see [14].

3 Concluding Remarks

We presented a general approach that expresses Data quality and cleaning tasks in terms of AI Planning problem, connecting two distinct research areas. Our approach has been formalised and fully-implemented on top of the UPMurphi planner, and applied to a real-life example analysing and cleaning million records concerning labour market movements of Italian citizens.

We are working on (i) including machine-learning algorithms to identify the *most suited* cleaning action, and (ii) applying our approach to build training sets for data cleaning tools based on machine-learning (e.g., [11]).

References

1. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications, Springer (2006)
2. Boselli, R., Cesarini, M., Mercorio, F., Mezzanzanica, M.: Planning meets data cleansing. In: The 24th ICAPS. AAAI Press (2014)
3. Chomicki, J.: Efficient checking of temporal integrity constraints using bounded history encoding. ACM Transactions on Database Systems (TODS) 20(2) (1995)
4. Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A.K., Ilyas, I.F., Ouzzani, M., Tang, N.: Nadeef: a commodity data cleaning system. In: SIGMOD (2013)
5. De Giacomo, G., Maggi, F.M., Marrella, A., Patrizi, F.: On the disruptive effectiveness of automated planning for LTLf-based trace alignment. In: AAAI (2017)
6. Della Penna, G., Intrigila, B., Magazzeni, D., Mercorio, F.: UPMurphi: a tool for universal planning on PDDL+ problems. In: The 19th ICAPS. pp. 106–113 (2009)
7. Della Penna, G., Intrigila, B., Magazzeni, D., Mercorio, F.: A PDDL+ benchmark problem: The batch chemical plant. In: ICAPS. pp. 222–224. AAAI Press (2010)
8. Della Penna, G., Magazzeni, D., Mercorio, F.: A universal planning system for hybrid domains. Applied Intelligence 36(4), 932–959 (2012)
9. Fan, W., Li, J., Ma, S., Tang, N., Yu, W.: Towards certain fixes with editing rules and master data. Proceedings of the VLDB Endowment 3(1-2), 173–184 (2010)
10. Hao, S., Tang, N., Li, G., He, J., Ta, N., Feng, J.: A novel cost-based model for data repairing. IEEE Transactions on Knowledge and Data Engineering 29(4) (2017)
11. Krishnan, S., Wang, J., Wu, E., Franklin, M.J., Goldberg, K.: Activeclean: Interactive data cleaning while learning convex loss models. arXiv preprint arXiv:1601.03797 (2016)
12. Mercorio, F.: Model checking for universal planning in deterministic and non-deterministic domains. AI Communications 26(2), 257–259 (2013)
13. Mezzanzanica, M., Boselli, R., Cesarini, M., Mercorio, F.: Data quality sensitivity analysis on aggregate indicators. In: DATA. pp. 97–108 (2012)
14. Mezzanzanica, M., Boselli, R., Cesarini, M., Mercorio, F.: A model-based approach for developing data cleansing solutions. The ACM Journal of Data and Information Quality 5(4), 1–28 (Mar 2015), http://doi.acm.org/10.1145/2641575
15. Mezzanzanica, M., Boselli, R., Cesarini, M., Mercorio, F.: A model-based evaluation of data quality activities in KDD. Information Processing & Management 51(2), 144–166 (2015)