Inductive Learning of Human Behaviours

Alessandra Russo*, Mark Law, and Krysia Broda

Department of Computing, Imperial College London,
{a.russo, mark.law09, k.broda}@imperial.ac.uk

Abstract. Over the last two decades there has been a growing interest in Inductive Logic Programming (ILP) [8], where the goal is to learn a logic program called a *hypothesis*, which together with a given background knowledge, explains a set of examples. The main advantage that ILP has over traditional statistical machine learning approaches is that the learned hypotheses can be easily expressed into plain English and explained to a human user, so facilitating a closer interaction between humans and machine. Although ILP has traditionally addressed the task of learning definite logic programs [9] (with no negation), our own recent ILP systems [2, 1, 4] have extended the field to learning under the answer set programming (ASP) semantics [3]. ASP programs are truly declarative. They allow additional constructs such as choice rules, hard and weak constraints, and support for non-monotonic inference. Choice rules and weak constraints are particularly useful for modelling human preferences, as the choice rules can represent the choices available to the user, and the weak constraints can specify which choices a human prefers. Our most recent system, ILASP [6], supports learning ASP programs with normal rules, choice rules, hard constraints and weak constraints.

1 The role of non-monotonic inductive learning

Non-monotonicity permits incremental learning, allowing the machine to periodically revise rules and knowledge learnt, as examples of user behaviours are continuously observed. The non-monotonicity property is particularly relevant in pervasive computing, where systems are expected to autonomously adapt to changes in user context and behaviour, whilst operating seamlessly with minimal user intervention. We have used our non-monotonic learning system, ASPAL [2], in mobile privacy [7], and enabled devices to learn and revise user's models from sensory input and user actions (e.g. user's actions on mobile devices). The learned models are accessed by mobile applications to determine automatic responses to events or requests, e.g., "if the user would allow access to his/her current location". The declarative representation of these models makes the system also capable of explaining its automatic responses to human users, and providing way for users to understand and amend what has been learnt.

* This research is partially funded by the EPSRC project EP/K033522/1 "Privacy Dynamics".

2 An example: learning human preferences

More recently, we have extended our ILASP [5] system[1] to learn weak constraints, which can be used to represent human-readable preferences. We used ILASP to learn route selection preferences from examples of which routes a user prefers to other routes [6]. A journey was encoded as a set of attributes of the legs of the journey; for example the journey $\{\texttt{distance}(\texttt{leg}(1), 2000), \texttt{distance}(\texttt{leg}(2), 100), \texttt{mode}(\texttt{leg}(1), \texttt{bus}), \texttt{mode}(\texttt{leg}(2), \texttt{walk})\}$ has two legs; in the first leg, the person must take a bus for `2000m` and in the second, he/she must walk `100m`.

For example, consider the following set of weak constraints:

$$WS = \begin{cases} :\sim \texttt{mode}(\texttt{L}, \texttt{walk}), \texttt{crime\_rating}(\texttt{L}, \texttt{R}), \texttt{R} > 3.[1@3, \texttt{L}, \texttt{R}] \\ :\sim \texttt{mode}(\texttt{L}, \texttt{bus}).[1@2, \texttt{L}] \\ :\sim \texttt{mode}(\texttt{L}, \texttt{walk}), \texttt{distance}(\texttt{L}, \texttt{D}).[\texttt{D}@1, \texttt{L}, \texttt{D}] \end{cases}$$

Using synthetically generated examples of partially ordered journeys, ILASP is able to learn the weak constraints WS, which reflect the *human preferences*:

1. `The user would like to avoid walking through an area with a high crime rating;`
2. `The user would like to minimise the number of buses taken;`
3. `The user would like to minimise the total distance walked.`

Note that ILASP also learns the priorities of the preferences (@3, @2 and @1 for the weak constraints in WS). These priorities indicate which preferences are considered to be more important by the user, with higher numbers reflecting higher importance.

Figure 1 shows the average accuracy of ILASP2i (our most recent system) on over 1000 journeys, with varying numbers of training examples. The average accuracy measures the proportion of pairs of journeys that ILASP2i orders correctly. With only 40 training examples of pairs $\langle j_1, j_2 \rangle$ such that j_1 is prefered to j_2, ILASP2i averages over 88%. This average accuracy increases to 93% when some of the training example pairs are such that j_1 is equally prefered to j_2.

3 Closing Remarks

These recent theoretical and practical advances in ILP demonstrate how machines can be empowered with human-like reasoning and learning abilities needed to maintain collaboration and communication with human users. At the same time their ability to combine declarative and optimisation inference within the learning process, may open up opportunities for exploring new ways of integrating quantitative and symbolic approaches to machine learning.

[1] For instructions on how to download and use ILASP see `http://www.doc.ic.ac.uk/~ml1909/ILASP`

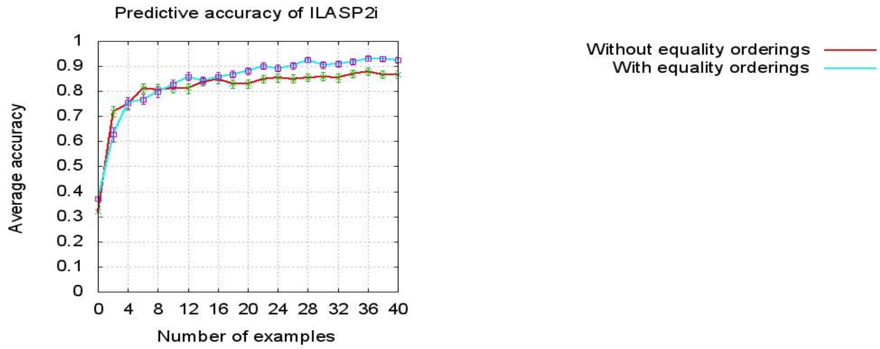

Fig. 1: average accuracy of ILASP2i

References

1. Athakravi, D., Corapi, D., Broda, K., Russo, A.: Learning through hypothesis refinement using answer set programming. In: Inductive Logic Programming, pp. 31–46. Springer (2014)
2. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming in answer set programming. In: Inductive Logic Programming, pp. 91–97. Springer (2012)
3. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP. vol. 88, pp. 1070–1080 (1988)
4. Law, M., Russo, A., Broda, K.: Inductive learning of answer set programs. In: Logics in Artificial Intelligence (JELIA 2014). Springer (2014)
5. Law, M., Russo, A., Broda, K.: Learning weak constraints in answer set programming. Theory and Practice of Logic Programming 15(4-5), 511–525 (2015)
6. Law, M., Russo, A., Broda, K.: Iterative learning of answer set programs from context-dependent examples. Theory and Practice of Logic Programming (2016)
7. Markitanis, A., Corapi, D., Russo, A., Lupu, E.: Learning user behaviours in real mobile domains. In: Latest Advances in Inductive Logic Programming. Imperial College Press (ILP 2011 Post-proceeding)
8. Muggleton, S.: Inductive logic programming. New generation computing 8(4), 295–318 (1991)
9. Muggleton, S.: Inverse entailment and progol. New generation computing 13(3-4), 245–286 (1995)