Worksheet #3
April 30th, 2020

Paper: ILP turns 20

- Given the tables in Fig. 1:

  1. explain the differences between these two tables. Why one is more suitable to traditional learning algorithms and the other is more suitable to relational learning?

     Table 1(a) shows a traditional input for data analysis, data mining and machine learning algorithms and systems. Rare exceptions accept tables with varying width size (association rules is an example). The second Table (1(b)) connects (relates) objects/individuals and values through properties.

  2. try representing each one of those tables using a relational approach.

     For Table 1(a), one very much used approach is to use binary predicates, where the key (in order to convert a key is needed) is used to index all columns which will become predicate names. For example, `author(e1,known)`, `thread(e1,new)`, `author(e2,unknown)`, etc. Table 1(b) yields directly predicates (in this case, binary, but not always): `likes(joe,resort_14)`, `type(resort_14,resort)`, `dislikes(joe,resort_35)`, etc.

  3. what kind of concepts could you learn from each one of these tables?

     Knowledge represented through relations can be queried and inference mechanisms are used to produce conclusions. However, new knowledge represented in the same way (in the same language) can be learned: new relations among already represented relations (hierachies), order relations (events that happen before or after another event, for example), same location relations etc.

Paper: ILP: theory and methods

1. What are the main limitations of classical machine learning models and algorithms?

   The first one, mentioned in the previous questions, is the limitation in the input format. A second one is not being able to allow learning among elements both row-wise and column-wise. A third limitation is that new knowledge can only be added by inserting new columns to the initial table. A fourth limitation sometimes is lack of interpretability and explainability of the learned models.

2. Explain "Prior Satisfiability", "Posterior Satisfiability", "Prior Necessity" and "Posterior Sufficiency". Why are these properties important? Can we violate or remove any one of these conditions to learn concepts? If so, why and in what conditions?

   - Prior Necessity: the background knowledge alone can not contain only positive examples.
   - Posterior Satisfiability: the hypothesis together with the background knowledge and negative examples need to prove something.
   - Prior Satisfiability: the background knowledge and the negative examples need to have something provable.
   - Posterior Sufficiency: H and B needs to prove some positive example.

3. What is the function of the "Absorption" inference rule in Inductive Logic Programming?

   To allow the generalization of a rule.

4. What is "θ-subsumption"

   It is an inference rule that, given a substitution θ and a rule $c1$, can allow a generalization $c2$, such that $c1\theta \subseteq c2$.

5. What is inverse resolution?

   Roughly speaking, inverse resolution is the opposite of deduction. In deduction, we take two clauses and simplify them producing a third one (if we have p -¿ q and q -¿ r then we can deduce p -¿ r or, simpler: if we have p and p -¿ q, then we can conclude q.) In inverse resolution, we take two facts and produce (invent/generalize) a rule that does not exist yet.

Paper: Turning 30: new ideas in ILP

1. Why have researchers started to look for other ways of using ILP to learn concepts? (most approaches used Prolog as a base language)

   Mainly, because ILP involves a machinery that, relying in Prolog engines, can become very inefficient, specially when larger amounts of relational data need to be processed. (the paper explains in more detail some of the issues related with limitations: lack of generalization for a larger number fo examples, need for positive ad negative cases, etc, but all of these can be mitigated and there are implementations that handle those well.

2. How can ILP concepts be learned using neural networks?

   There are at least three main paths: (1) take relational knowledge, find a suitable way of encoding it and feed it to a neural network, (2) learn rules using ILP, represent them as new features in a bidimensional table (they become boolean variables), find an appropriate encoding and feed it to a neural network, (3) use the relational knowledge to build the structure of the network.

3. What is predicate invention?

   Illustration with an example: it is in the paper, page 3, Section 3.1.