

## Short detour:- SRL

One of the central open questions of artificial intelligence is concerned with combining (i) expressive knowledge representation formalisms such as relational and first-order logic with (ii) principled probabilistic and statistical approaches to inference and learning. Why? Here are some reasons:

1. The fields of knowledge representation and inductive logic programming stress the importance of relational and logical representations that provide the flexibility and modularity to model large domains. They also highlight the importance of making general statements, rather than making statements for every single aspect of the world separately.
2. The fields of statistical learning and uncertainty in artificial intelligence emphasize that agents that operate in the real world must deal with uncertainty. An agent typically receives only noisy or limited information about the world; actions are often non-deterministic; and an agent has to take care of unpredictable events. Probability theory provides a sound mathematical foundation for inference and learning under uncertainty.
3. Machine learning, in general, argues that an agent needs to be capable of improving its performance through experience.

} Logical inference comes handy

} Traditionally for attribute-value data [Matrix]

→ Brings ① & ② together in an application.  
→ Because real world applications are complex & heterogeneous

**Definition 19** Statistical relational learning deals with machine learning and data mining in relational domains where observations may be missing, partially observed, and/or noisy.

# Why logical abstraction in SRL?

Employing relational and logical abstraction within statistical learning has three advantages:

1. Variables, i.e., placeholders for entities allow one to make abstraction of specific entities.
2. Unification allows one to share information among entities.
3. In many applications, there is a rich background theory available, which can efficiently and elegantly be represented as sets of general regularities.

} Instead of finding regularities for each entity independently, find regularities for "entity groups"  
Gives declarative & compact knowledge

- ↓
- ① restricts search space for concept
  - ② Allows reuse of experience

$$\Sigma = \mathcal{H} \cup \mathcal{B} \cup \dots$$

↓  
learn.

Given:  $E^+ E^-$

slightly diff semantics than multiclass.

① WordNet analogy

② Nearest neighbor analogy { Needs to be recombined at run time using inference

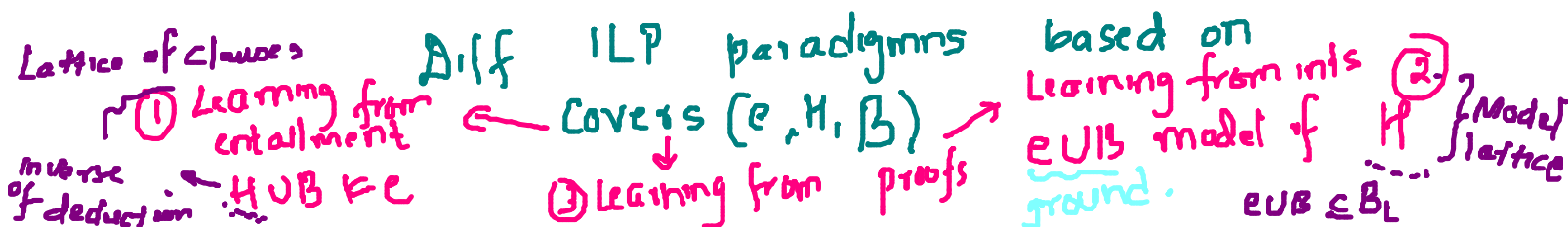
ILP:- A naive approach to SRL  
 [Multi Relational data mining]

↓  
 ① Tries to learn logic programs ( $\Sigma$ ): Eg: Horn clauses

② But does not explicitly handle noise

$\Sigma \equiv$  ILP setting

**Definition 20** Given a set of positive and negative examples  $\mathcal{E}^+$  and  $\mathcal{E}^-$  over some language  $\mathcal{L}_E$ , a background theory  $B$ , in the form of a set of definite clauses, a hypothesis language  $\mathcal{L}_H$ , which specifies the clauses that are allowed in hypotheses, and a covers relation  $\text{covers}(e, H, B) \in \{0, 1\}$ , which basically returns the classification of an example  $e$  with respect to  $\mathcal{H}$  and  $B$ , find a hypothesis  $H$  in  $\mathcal{H}$  that covers (with respect to the background theory  $B$ ) all positive examples in  $\mathcal{E}^+$  (completeness) and none of the negative examples in  $\mathcal{E}^-$  (consistency).



Example : Q: Does an ILP system delve into the semantics of pred symbols?  
No!

Formally, ILP is concerned with finding a hypothesis  $H$  (a logic program, *i.e.* a definite clause program) from a set of positive and negative examples  $\mathcal{E}^+$  and  $\mathcal{E}^-$ .

Consider learning a definition for the *Daughter/2* predicate, *i.e.*, a set of clauses with head predicates over *Daughter/2*, given the following facts as learning examples:

$\mathcal{E}^+$ :	<i>Daughter(dorothy, ann).</i>	} pos
	<i>Daughter(dorothy, brian).</i>	
<hr/>		
$\mathcal{E}^-$ :	<i>Daughter(rex, ann).</i>	} neg
	<i>Daughter(rex, brian).</i>	

Additionally, we have some general knowledge called background knowledge  $\mathcal{B}$ , which describes the family relationships and sex of each person:

<i>Mother(ann, dorothy).</i>	<i>F<sup>f</sup>emale(dorothy).</i>	<i>F<sup>f</sup>emale(ann).</i>
<i>Mother(ann, rex).</i>	<i>F<sup>f</sup>ather(brian, dorothy).</i>	<i>F<sup>f</sup>ather(brian, rex).</i>

From this information, we could induce the following  $H$ :

D

<i>Daughter(C, P)</i>	:	-	<i>F<sup>f</sup>emale(C), Mother(P, C).</i>
<i>Daughter(C, P)</i>	:	-	<i>F<sup>f</sup>emale(C), F<sup>f</sup>ather(P, C).</i>

D

which perfectly explains the examples in terms of the background knowledge, *i.e.*,  $\mathcal{E}^+$  are entailed by  $H$  together with  $\mathcal{B}$ , but  $\mathcal{E}^-$  are not entailed.

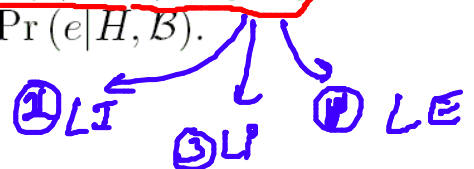
→ Ted to common sense knowledge

# Benefits of SRL based on ILP

1. Classical ILP learning settings, as we will argue, naturally carry over to the probabilistic case. The probabilistic ILP settings make abstraction of specific probabilistic relational and first order logical representations and inference and learning algorithms yielding general statistical relational learning settings.
2. Many ILP concepts and techniques such as *more-general-than*, *refinement operators*, *least general generalization* (lub), and *greatest lower bound* (glb) can be reused. Therefore, many ILP learning algorithms such as Quinlan's FOIL and De Raedt and Dehaspes' Claudien can easily be adapted.
3. The ILP perspective highlights the importance of background knowledge within statistical relational learning. The research on ILP and on artificial intelligence in general has shown that background knowledge is the key to success in many applications.
4. An ILP approach should make statistical relational learning more intuitive to those coming from an ILP background and should cross-fertilize ideas developed in ILP and statistical learning.

# Probabilistic ILP Setting

2 changes:-

1. Probabilistic Clauses: (Clauses in  $H$  and  $\mathcal{B}$  are annotated with probabilistic information, and Here, we use the following probability notations.) With  $X$ , we denote a (random) variable. Furthermore,  $x$  denotes a state and  $\mathbf{X}$  (resp.  $\mathbf{x}$ ) a set of variables (resp. states). We will use  $\text{Pr}$  to denote a probability distribution, e.g.,  $\text{Pr}(x)$ , and  $p$  to denote a probability value, e.g.,  $p(X = x)$  and  $p(X = \mathbf{x})$ .
  2. Probabilistic Covers: (The covers relation becomes probabilistic.) A probabilistic covers relation softens the hard covers relation employed in traditional ILP and is defined as the probability of an example given the hypothesis and the background theory. A probabilistic covers relation takes as arguments an example  $e$ , a hypothesis  $H$  and possibly the background theory  $\mathcal{B}$ , and returns the probability value  $\text{Pr}(e|H, \mathcal{B}) \in [0, 1]$  of the example  $e$  given  $H$  and  $\mathcal{B}$ , i.e.,  $\text{covers}(e, H, \mathcal{B}) = \text{Pr}(e|H, \mathcal{B})$ .
- 

# FORMAL DEFINITION

(Markov Networks) :  $P(\bar{x}) = \prod_{x_f} P(x_f)$

**Definition 21** Given a probabilistic-logical language  $\mathcal{L}_H$  and a set  $\mathcal{E}$  of examples over some language  $\mathcal{L}_E$ , find the hypothesis  $H^*$  in  $\mathcal{L}_H$  that maximizes  $\Pr(\mathcal{E}|H^*, \mathcal{B})$ .

Not prob.

Prob reward.

Under the usual *i.i.d.* assumption, *i.e.*, examples are sampled independently from identical distributions, this results in the maximization of

$\mathcal{L}_H$  is the hypothesis

$$\Pr(\mathcal{E}|H^*, \mathcal{B}) = \prod_{e \in \mathcal{E}} P(e|H^*, \mathcal{B}) = \prod_{e \in \mathcal{E}} \text{covers}(e, H^*, \mathcal{B})$$

optimal hypothesis

How you decompose this depends on  $\mathcal{L}_H$ .

- ①  $\mathcal{L}_H$     ②  $\mathcal{L}_E$ ,    ③  $\Pr(\mathcal{E}|H^*, \mathcal{B})$ .