Probabilistic Inductive Constraint Logic

Fabrizio Riguzzi[1] Elena Bellodi[2] Riccardo Zese[2]
Giuseppe Cota[2] Evelina Lamma[2]

Dipartimento di Matematica e Informatica – University of Ferrara

Dipartimento di Ingegneria – University of Ferrara
[fabrizio.riguzzi,elena.bellodi,evelina.lamma,
riccardo.zese,giuseppe.cota]@unife.it

ILP 2015

Probabilistic Logics

- Probabilistic logic models have successful application in a variety of fields
- However, inference and learning is expensive
- Proposals such as Tractable Markov Logic [Domingos, Webb, AAAI 2012], Tractable Probabilistic Knowledge Bases [Webb, Domingos, StarAI 2013][Niepert, Domingos, StarAI 2014] and fragments of probabilistic logics [van den Broeck, NIPS 2011][Niepert, van den Broeck, AAAI 2014] strive to achieve tractability by limiting the form of sentences.
- In ILP, the learning from interpretation settings [De Raedt, Dzeroski, AI 1994][Blockeel et al, 1999] offers advantages in terms of tractability: learning first-order clausal theories is tractable [De Raedt, Dzeroski, AI 1994], examples in learning from interpretations can be considered in isolation [Blockeel et al, 1999]

Objectives

- Inductive Constraint Logic (ICL) [De Raedt, Van Laer, ALT 1995]: performs discriminative learning from interpretations
- Models are sets of integrity constraints
- We want to consider a probabilistic version of the sets of integrity constraints with a semantics in the style of the distribution semantics [Sato, ICLP 1995]
- Each integrity constraint is annotated with a probability and a model assigns a probability of being positive to interpretations
- This probability can be computed in linear time given the number of groundings of the constraints.

ICL

- ICL [De Raedt, Van Laer, ALT 1995] performs discriminative learning from interpretations
- Constraint Logic Theory: a set of Integrity Constraints of the form

$$L_1, \ldots, L_b \rightarrow A_1; \ldots; A_h \tag{1}$$

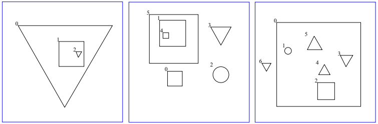
  B: a background knowledge
- A CLT T classifies an interpretation I as positive given a background knowledge B if $M(B \cup I) \models T$
- *range-restricted* clause: all the variables that appear in the head also appear in the body.
- If T is range-restricted, $M(B \cup I) \models T$ can be tested by asking the goal

$$?- Body(C), \neg Head(C).$$

  against a Prolog database containing I and B. If the query fails, C is true in I given B, otherwise C is false in I given B.

Example: Bongard Problems

- Discriminate between positive and negative pictures containg geometric shapes.



- Each picture can be described by an interpretation

$$I_l = \{triangle(0), large(0), square(1), small(1), inside(1,0), \quad (2)$$
$$triangle(2), inside(2,1)\} \quad (3)$$

- $B = \begin{array}{ll} in(A,B) & \leftarrow inside(A,B). \\ in(A,D) & \leftarrow inside(A,C), in(C,D). \end{array}$
- $M(B \cup I_l) \supseteq \{in(1,0), in(2,1), in(2,0)\}$
- $C_1 = triangle(T), square(S), in(T,S) \rightarrow false$ is false in I_l given B
- In the central picture instead C_1 is true given B

ICL

- ICL uses a covering loop on the negative examples
- It starts from an empty theory and adds one IC at a time
- After the addition of the IC, the set of negative examples that are ruled out by the IC are removed from the overall set of negative examples
- The loop ends when no more ICs can be generated or when the set of negative examples becomes empty
- The IC to be added is found by beam search with $P(\ominus|\overline{C})$ as the heuristic function (the precision on negative examples)

Probabilistic Constraint Logic

- A Probabilistic Constraint Logic Theory (PCLT) is a set of probabilistic integrity constraints (PICs)

$$p_i \;::\; L_1, \ldots, L_b \rightarrow A_1; \ldots; A_h \tag{4}$$

- A PCLT T defines a probability distribution on ground constraint logic theories called worlds: for each grounding of each IC, we include the IC in a world with probability p_i and we assume all groundings to be independent
- Constraint C_i has n_i groundings called $C_{i1}, \ldots, C_{in_i}$.
- The probability of a world w is given by the product:

$$P(w) = \prod_{i=1}^{n} \prod_{C_{ij} \in w} p_i \prod_{C_{ij} \notin w} (1 - p_i).$$

Probabilistic Constraint Logic

- The probability $P(\oplus|w, I)$ of the positive class given an interpretation I, a background knowledge B and a world w is 1 if $M(B \cup I) \models w$ and 0 otherwise.
- The probability $P(\oplus|I)$ of the positive class given an interpretation I and a background B is the probability of a PCLT T satisfying I
- $P(\oplus|I)$ is given by

$$P(\oplus|I) = \sum_{w \in W} P(\oplus, w|I) = \sum_{w \in W} P(\oplus|w, I)P(w|I) = \quad (5)$$

$$\sum_{w \in W, M(B \cup I) \models w} P(w) \quad (6)$$

$P(\ominus|I) = 1 - P(\oplus|I)$.

Probabilistic Constraint Logic

- There is an exponential number of worlds
- We can associate a Boolean random variable X_{ij} to each instantiated constraint C_{ij}. Let $\mathbf{X}$ be the set of the X_{ij} variables. These variables are all mutually independent
- We must keep only the worlds where $\overline{X_{ij}}$ holds for all ground constraints C_{ij} violated in I.
- I satisfies all the worlds where the formula

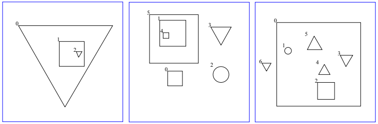
$$\phi = \bigwedge_{i=1}^{n} \bigwedge_{M(B \cup I) \not\models C_{ij}} \overline{X_{ij}}$$

is true

$$P(\oplus|I) = P(\phi) = \prod_{i=1}^{n}(1 - p_i)^{m_i} \qquad (7)$$

where m_i is the number of instantiations of C_i that are not satisfied in I

Example: Bongard Problems



- Consider the PCLT
  $\{C_1 = 0.5 \ :: \ triangle(T), square(S), in(T, S) \rightarrow false\}$
- In the left picture the body of C_1 is true for the single substitution $T/2$ and $S/1$ thus $m_1 = 1$ and $P(\oplus|I_l) = 0.5$.
- In the right picture the body of C_1 is true for three couples (triangle, square) thus $m_1 = 3$ and $P(\oplus|I_r) = 0.125$.

Learning Probabilistic Constraint Logic Theories

Given

- a set $\mathcal{I}^+ = \{I_1, \ldots, I_Q\}$ of positive interpretations
- a set $\mathcal{I}^- = \{I_{Q+1}, \ldots, I_R\}$ of negative interpretations
- a normal logic program *B* (background knowledge)

Find: a PCLT *T* such that the likelihood

$$L = \prod_{q=1}^{Q} P(\oplus|I_q) \prod_{r=Q+1}^{R} P(\ominus|I_r)$$

is maximized.

The likelihood can be unfolded to

$$L = \prod_{q=1}^{Q} \prod_{l=1}^{n} (1-p_l)^{m_{lq}} \prod_{r=Q+1}^{R} \left(1 - \prod_{l=1}^{n} (1-p_l)^{m_{lr}}\right) \quad (8)$$

where m_{iq} (m_{ir}) is the number of instantiations of C_i that are false in I_q (I_r) and n is the number of ICs.

Parameter Learning

- Let us compute the derivative of the likelihood with respect to the parameter p_i

$$\frac{\partial L}{\partial p_i} = \frac{L}{1-p_i}\left(\sum_{r=Q+1}^{R} m_{ir}\frac{P(\oplus|I_r)}{P(\ominus|I_r)} - m_{i+}\right) \quad (9)$$

- where $m_{i+} = \sum_{q=1}^{Q} m_{iq}$
- The equation $\frac{\partial L}{\partial p_i} = 0$ does not admit a closed form solution so we must use optimization to find the maximum of L
- We can optimize the likelihood with Limited-memory BFGS (L-BFGS) [Nocedal, MathComp 1980]
- L-BFGS requires the computation of L and of its derivative at various points.

Structure Learning

- First search for good candidate ICs, then search for a theory guided by the LL of the data
- Search for ICs: bottom-up beam search. The revisions are scored by the log likelihood (LL) resulting from parameter learning
- The refinement operator adds literals from a top IC obtained by saturation as in Progol using mode declarations
- A fixed-size list with the best ICs found so far is kept

Structure Learning

- Seach for a theory: greedy search in the space of theories by iteratively adding an IC *CI* from the list of best clauses ordered by *LL*
- The IC is kept if the log likelihood *LL* after parameter learning improves

Related Work

- Similarity with the distribution semantics
- Inference in the DS is #P in the number of variables
- On the contrary, computing the probability of the positive class given an interpretation in a PCLT is linear in the number of variables.
- 1BC [Flach, Lachiche, ML 2004] induces first-order features in the form of conjunctions of literals and combines them using naive Bayes in order to classify examples
- First-order features are similar to integrity constraints with an empty head
- The probability of a feature is computed by relative frequency in 1BC
- This can lead to suboptimal results if compared to PASCAL, where the probabilities are optimized to maximize the likelihood.

Experiments

- PASCAL has been implemented in SWI-Prolog
- For performing L-BFGS we ported the YAP-LBFGS library developed by Bernd Gutmann to SWI-Prolog. This library is based on libLBFGS.
- Hardware: machines with an Intel Xeon Haswell E5-2630 v3 (2.40GHz) CPU and 128 GB RAM
- Comparison with DPML [Lamma et al, ILP 2007] (similar to ICL)
- Process mining dataset [Bellodi et al, KSEM 2010]: careers of students enrolled at the University of Ferrara
- 776 interpretations each corresponding to a different student career
- Students who graduated: positive interpretations; student who did not finish their studies: negative interpretations

Experiments

- Five-fold cross validation

System	LL	AUCROC	AUCPR	Accuracy	Time(s)
PASCAL	-302.664	0.923	0.851	0.889	568.509
DPML	-440.254	0.707	0.53	0.656	280.594

Conclusions and Future Work

- Conclusions
  - Tractable inference
  - Parameter optimization by L-BFGS
  - Good initial results
- Future work
  - Test on more datasets
  - Distributed learning

THANKS FOR LISTENING

AND

ANY QUESTIONS ?